Vytěžování dat

Filip Železný

Katedra kybernetiky
skupina Inteligentní Datové Analýzy (IDA)

18. března 2009

Z minulé přednášky

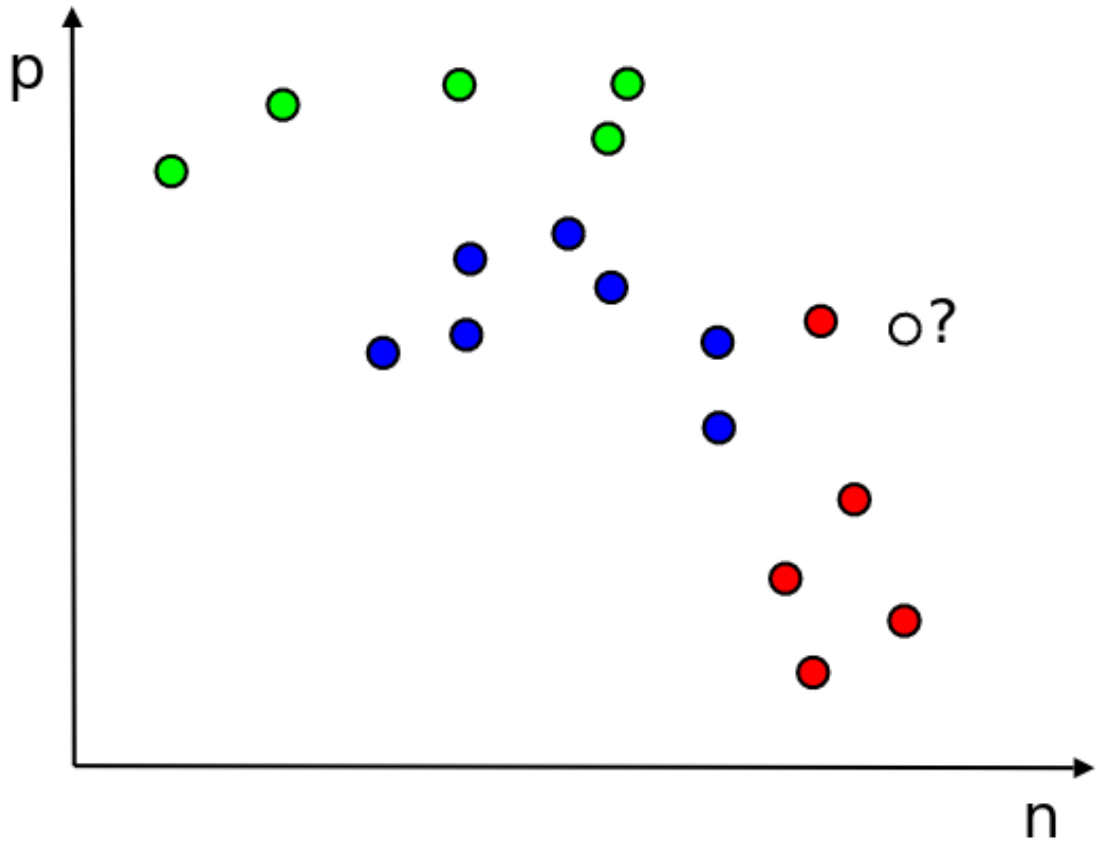
Připomínka klasifikačního příkladu

Příjmy (p)	**Rok narození (n)**	**Úvěr (u)**
vysoké	1969	splácí
nízké	1974	nesplácí
střední	1940	problémy
nízké	1985	problémy
...	...	...

Cílová veličina je **Úvěr**.

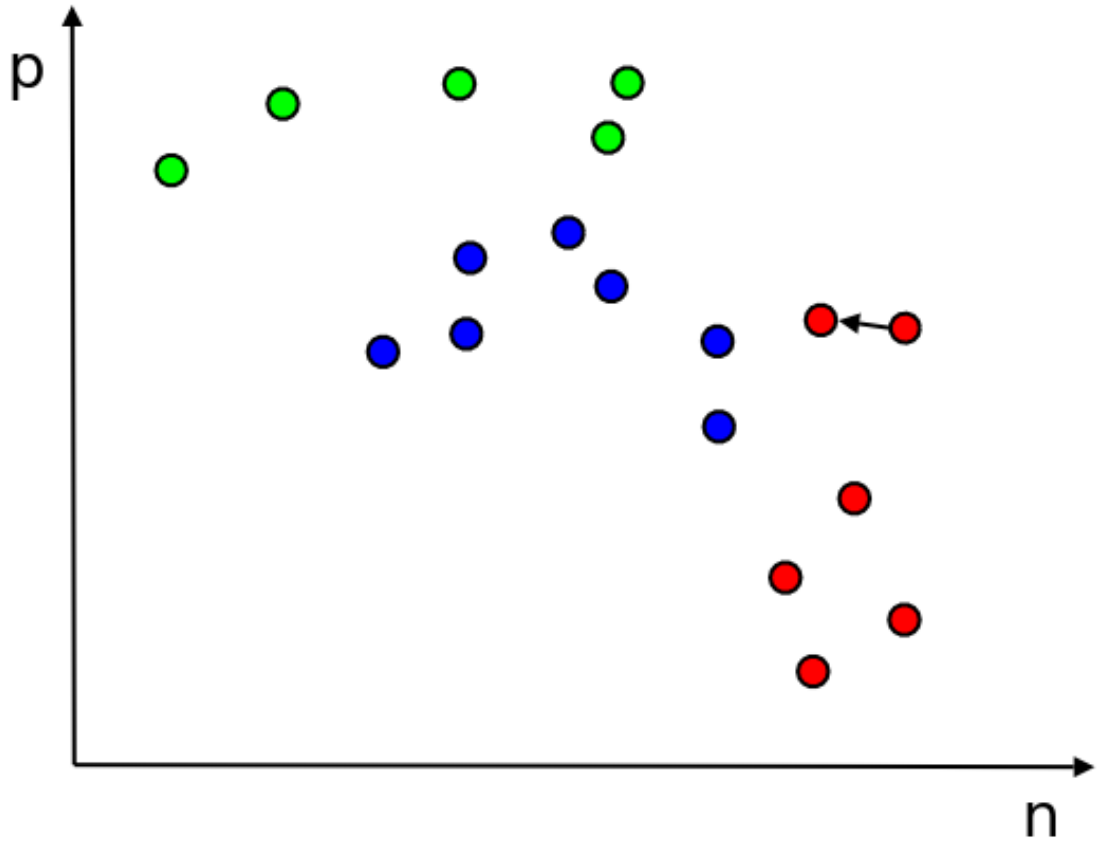
Klasifikace dle nejbližšího souseda

- Metoda, která nevyžaduje odhad pravděpodobností.
- Předpoklad: umíme spočítat podobnost dvou datových instancí
- Zde $p, n \in N$, $u \in \{$splácí, problémy, nesplácí$\}$



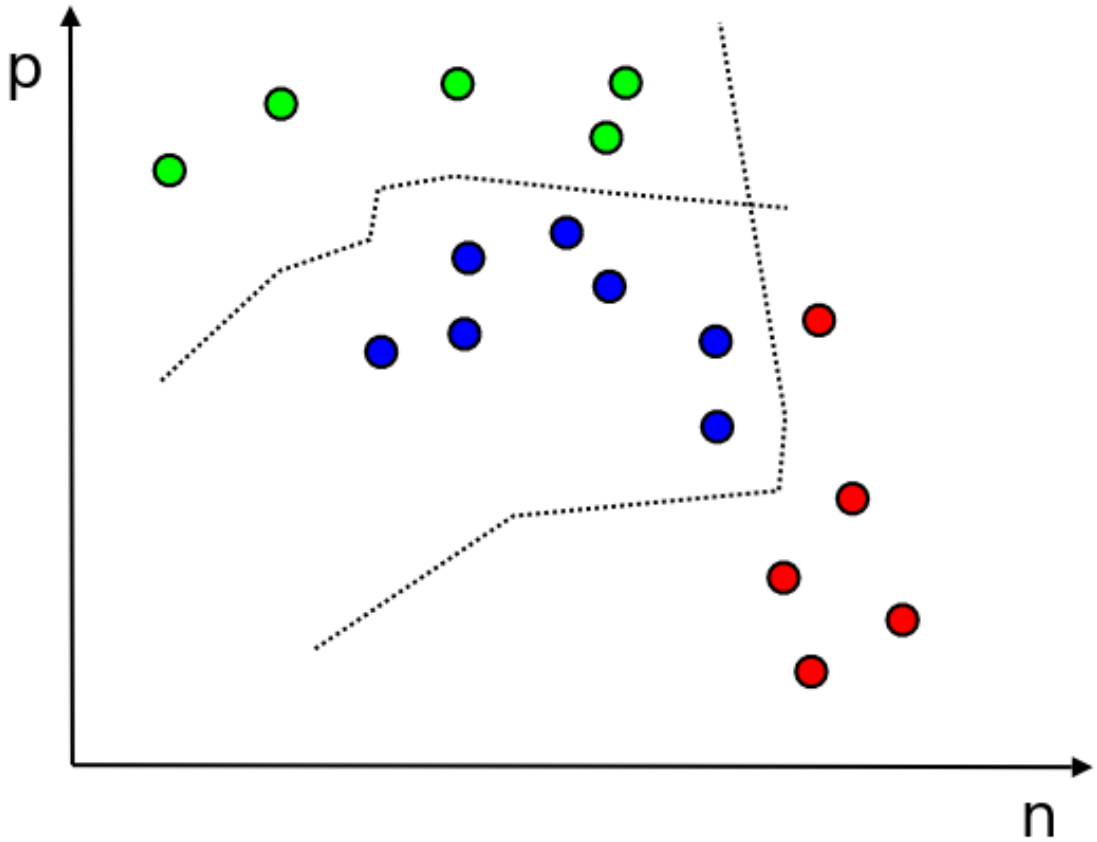
Klasifikace dle nejbližšího souseda

- Zde podobnost např. $(p_1 - p_2)^2 + (n_1 - n_2)^2$
- (Eukleidovský prostor).
- Zařazujeme do třídy nejpodobnější instance.



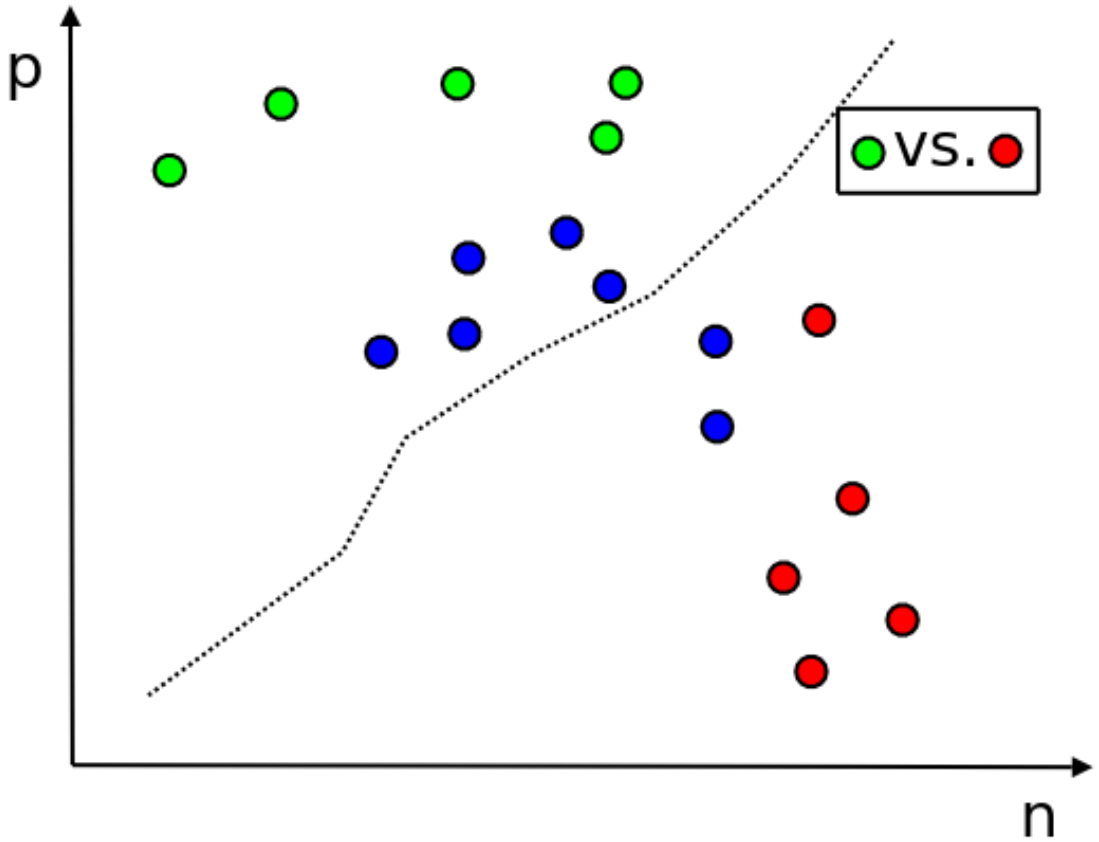
Separační křivky

Po částech lineární



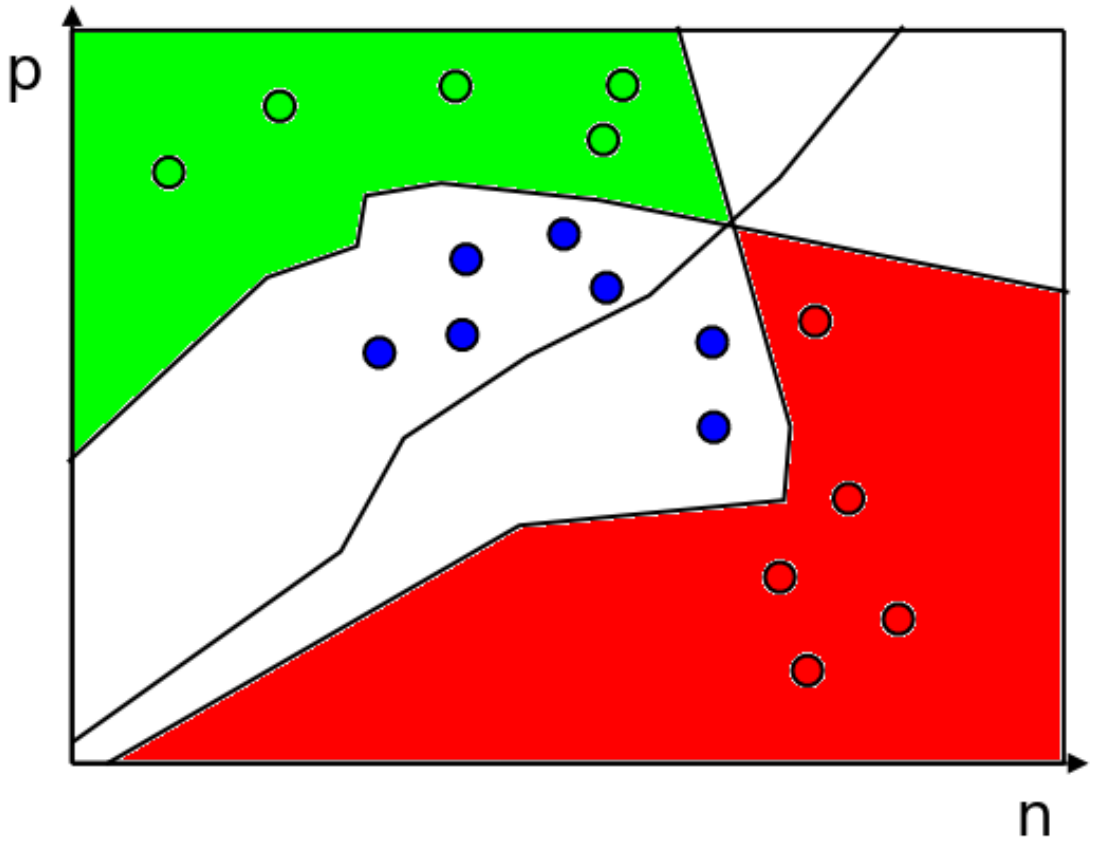
Separační křivky

Po částech lineární



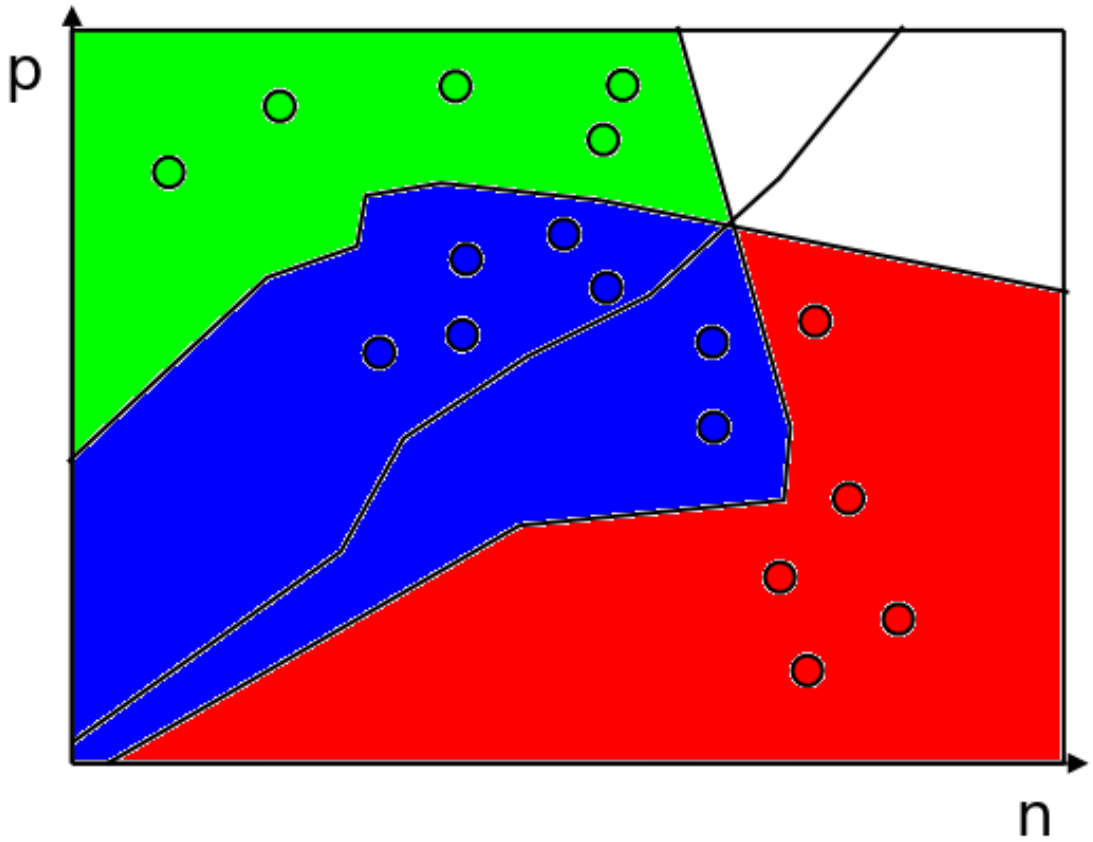
Separace tříd

Lineárně omezené oblasti



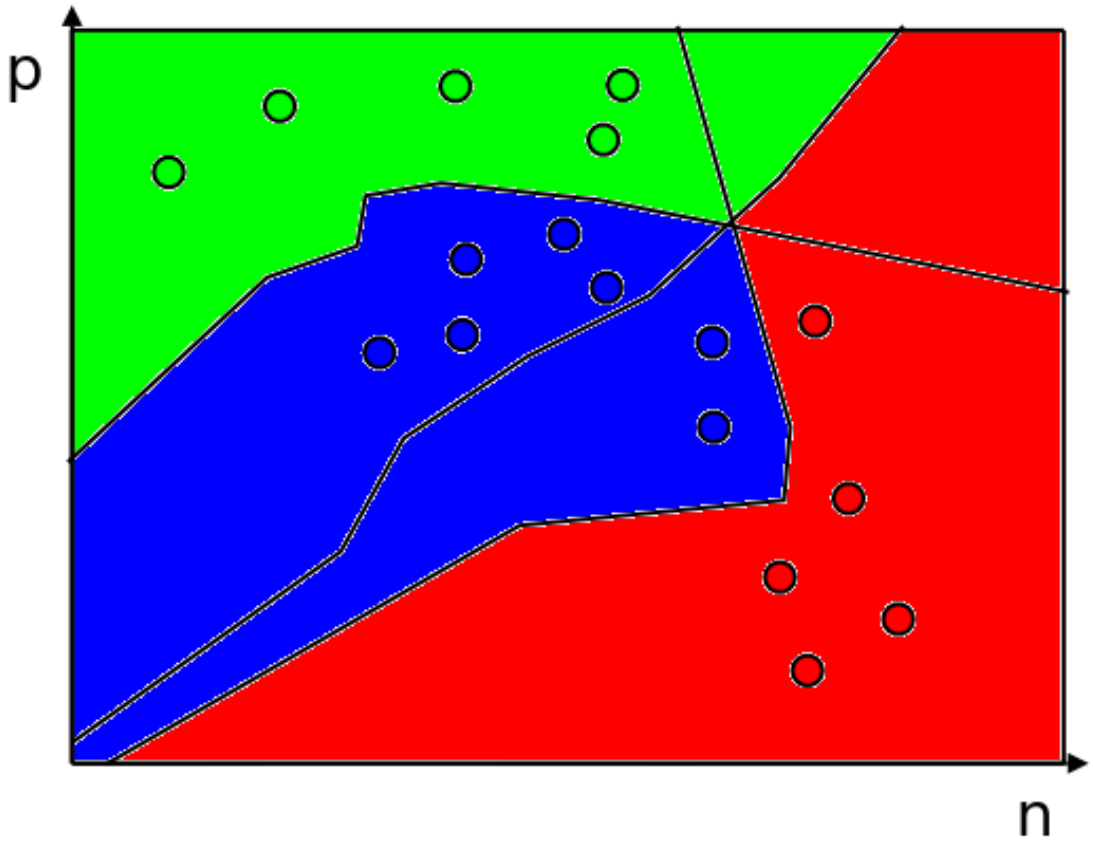
Separace tříd

Lineárně omezené oblasti



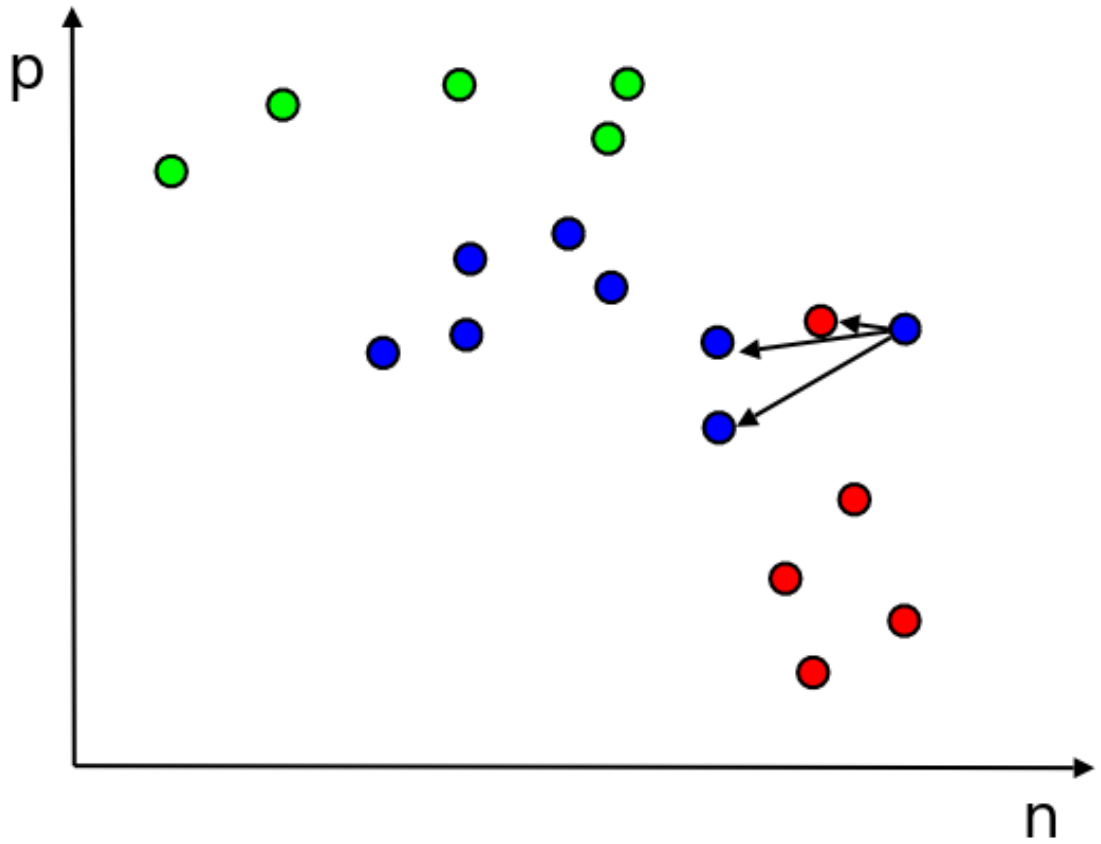
Separace tříd

Lineárně omezené oblasti



Klasifikace dle k nejbližších sousedů

Zařazujeme do třídy převládající mezi k nejpodobnějšími instancemi.



Trénovací a skutečná chyba klasifikátoru

Trénovací chyba

Podíl instancí v datech chybně klasifikovaných klasifikátorem sestrojeným z těchto dat (tzv. trénovacích dat).

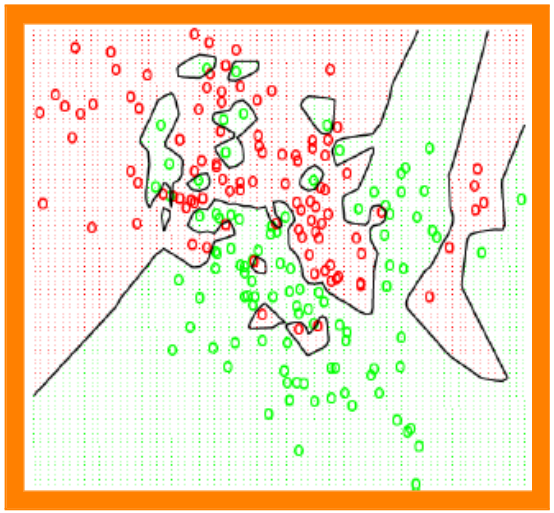
Skutečná chyba

Pravděpodobnost chybné klasifikace určená pravděpodobnostním rozdělením, z něhož jsou vybírána trénovací data. Totéž, co střední hodnota rizika při použití ztrátové funkce L_{01}:

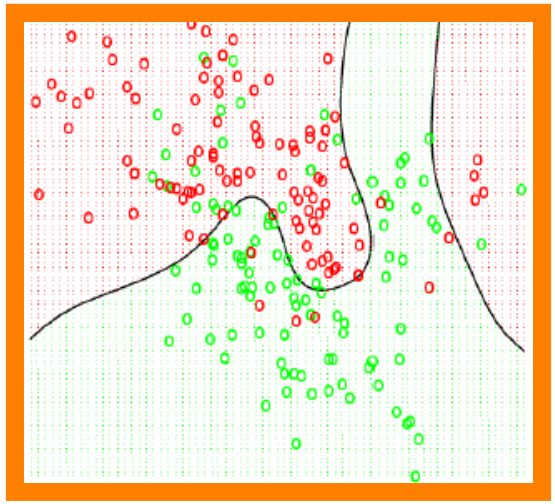
$$\sum_x r_{u|p}(y, x) \cdot P_p(x)$$

Jaké k je nejlepší?

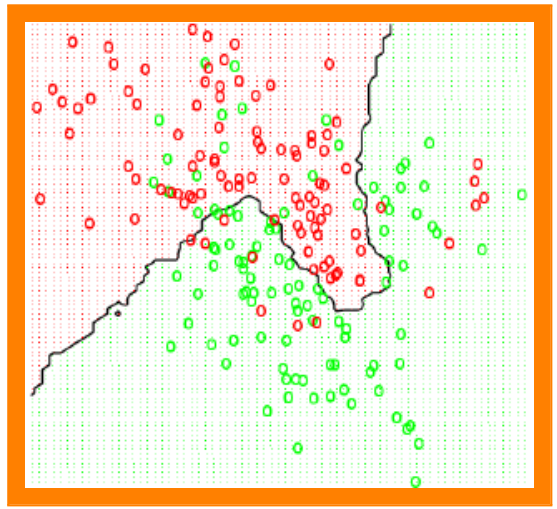
Experiment: generovaná data [Hastie et al.: Elements of Statistical Learning]



$k = 1$
Nulová trénovací chyba, přesto klasifikace odlišná od optimální→



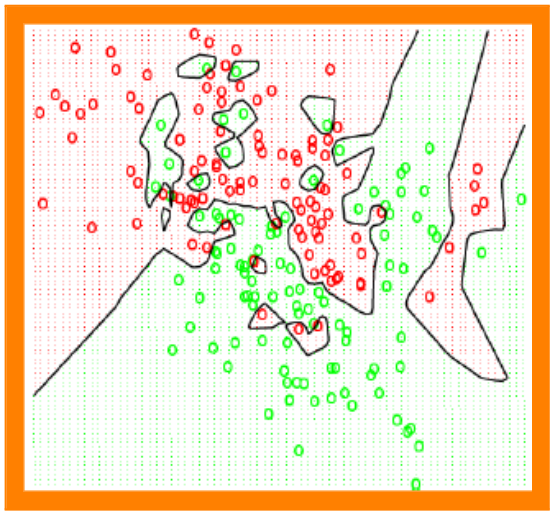
Dle maximální aposteriorní pravděpodobnosti (optimální)



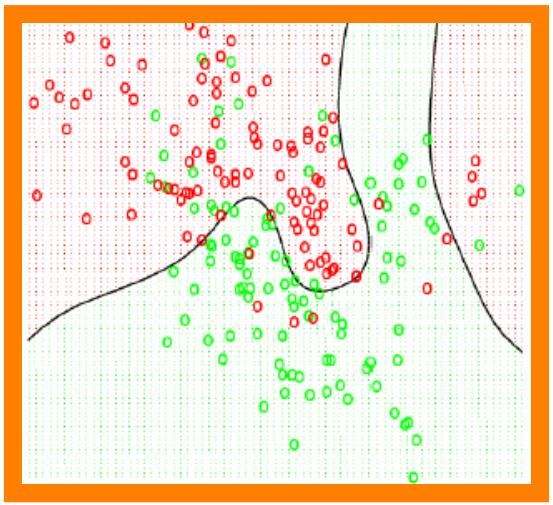
$k = 15$
Malá náchylnost k šumu, přesto klasifikace odlišná od ←optimální

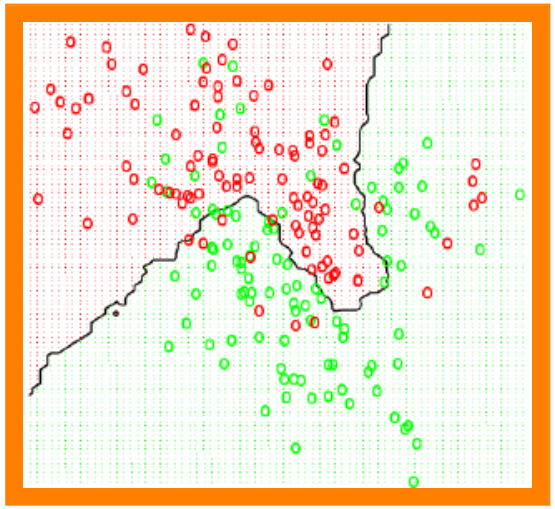
Jaké k je nejlepší?

Experiment: generovaná data [Hastie et al.: Elements of Statistical Learning]



Separace složitá, ,,kostrbatá"

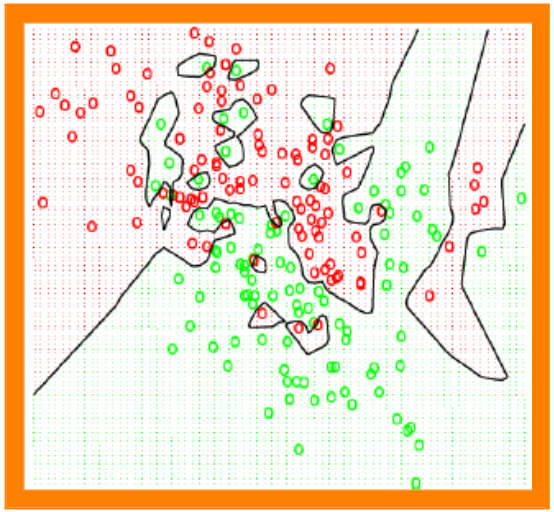




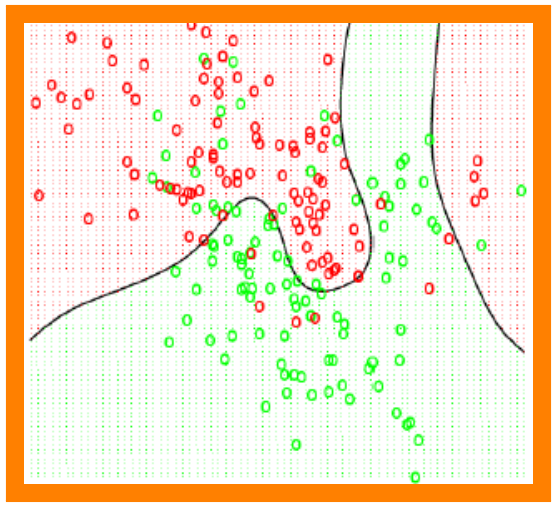
Separace jednoduchá, regulární

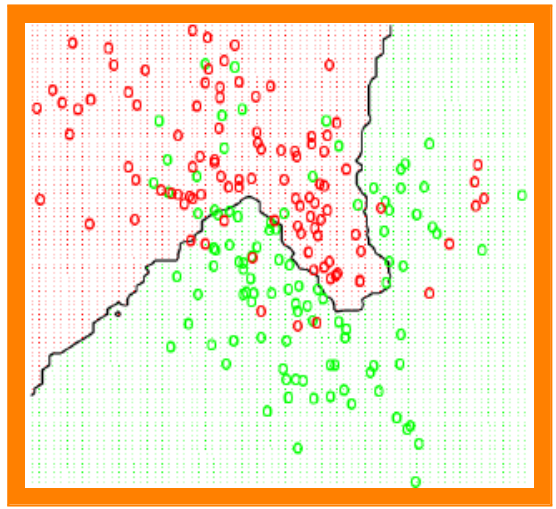
Jaké k je nejlepší?

Experiment: generovaná data [Hastie et al.: Elements of Statistical Learning]



Separace složitá, ,,kostrbatá"





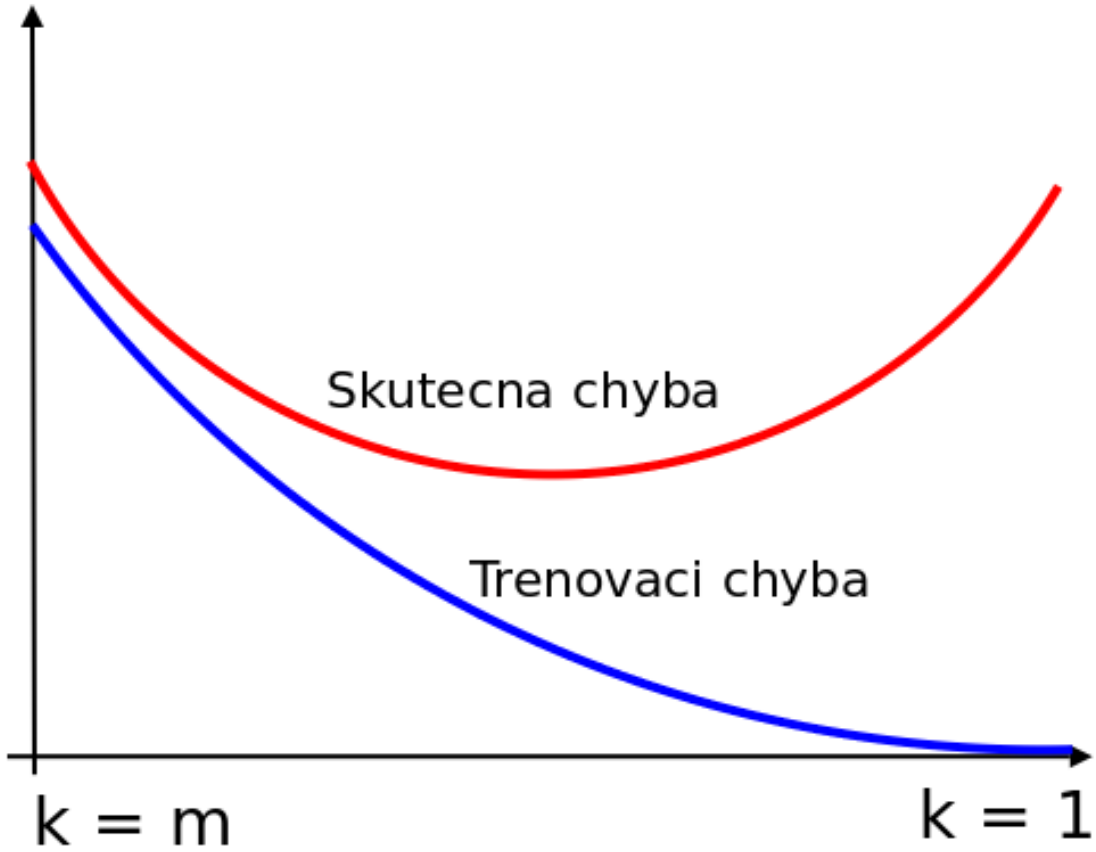
Separace jednoduchá, regulární

Povolíme-li složitou separaci,

- snadněji dosáhneme nízké trénovací chyby
- trénovací chyba méně vypovídá o skutečné chybě (,,přeučení")

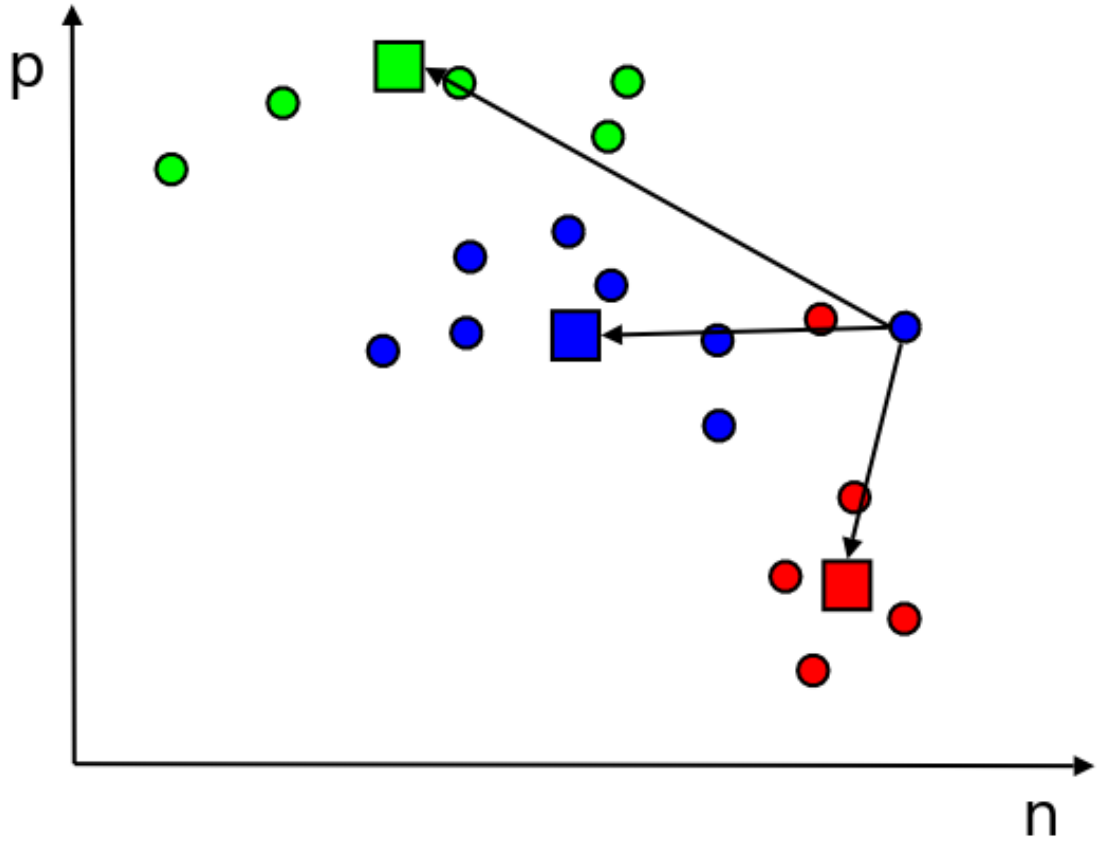
Trénovací vs. skutečná chyba

- Typický průběh chyb pro $k = m$ (počet dat), $m - 1, \ldots 1$
- Trénovací chyba není dobrým odhadem skutečné chyby!



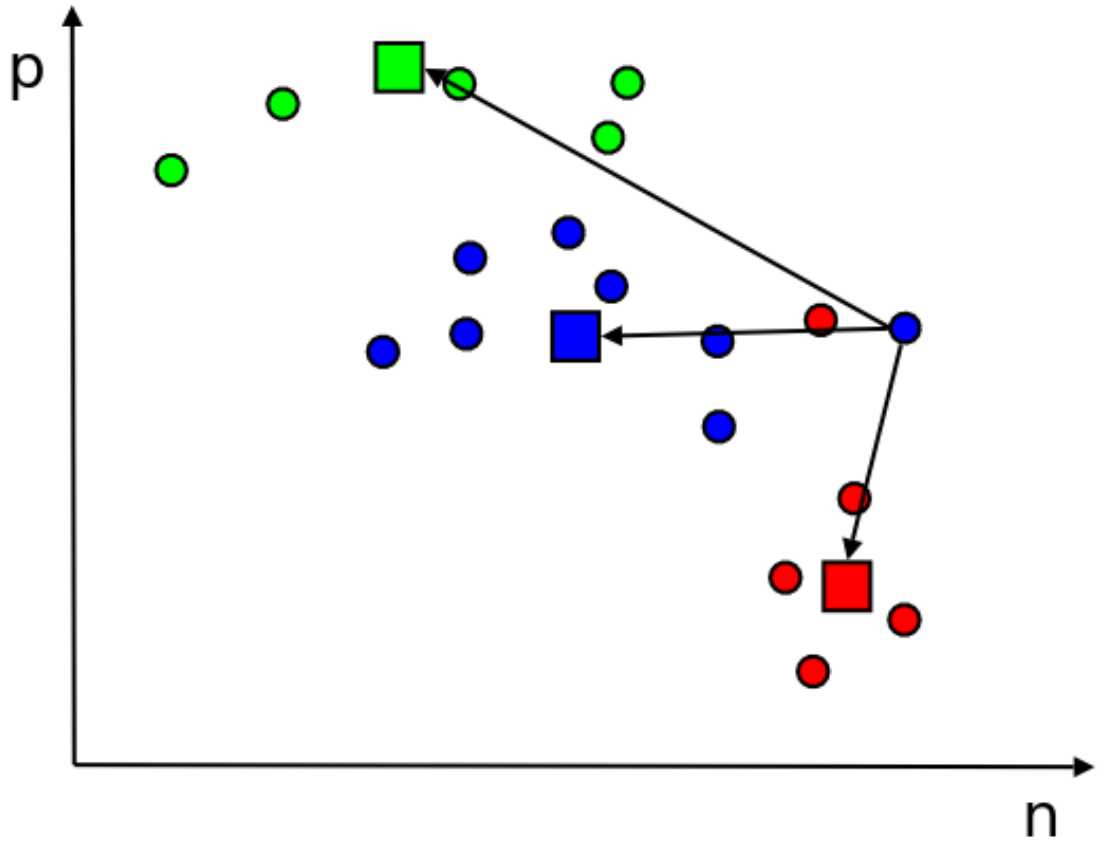
Klasifikace dle etalonů

- Metoda nevyžadující přístup ke všem instancím při klasifikaci.
- Pro každou třídu je definován **etalon**.



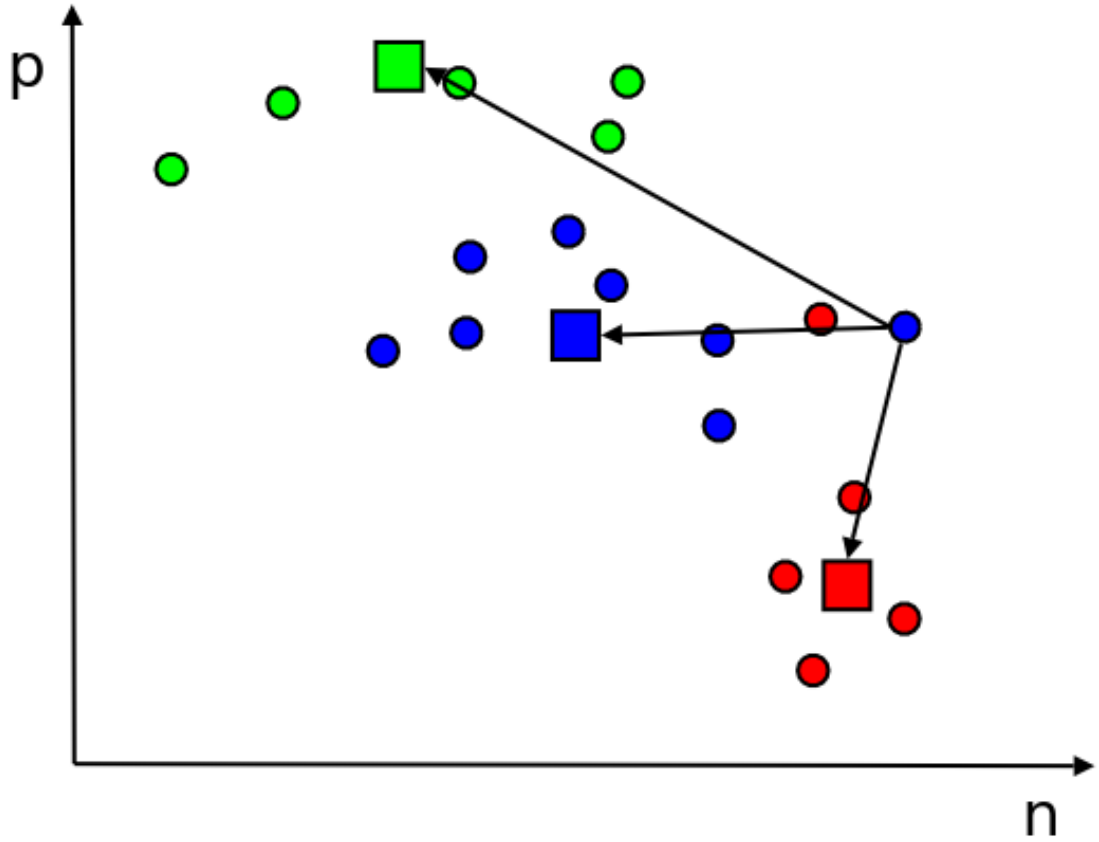
Klasifikace dle etalonů

- Etalon má minimální průměrnou vzdálenost k instancím třídy.
- Vybírán z trénovacích dat, nebo z celého oboru hodnot (zde p, n).

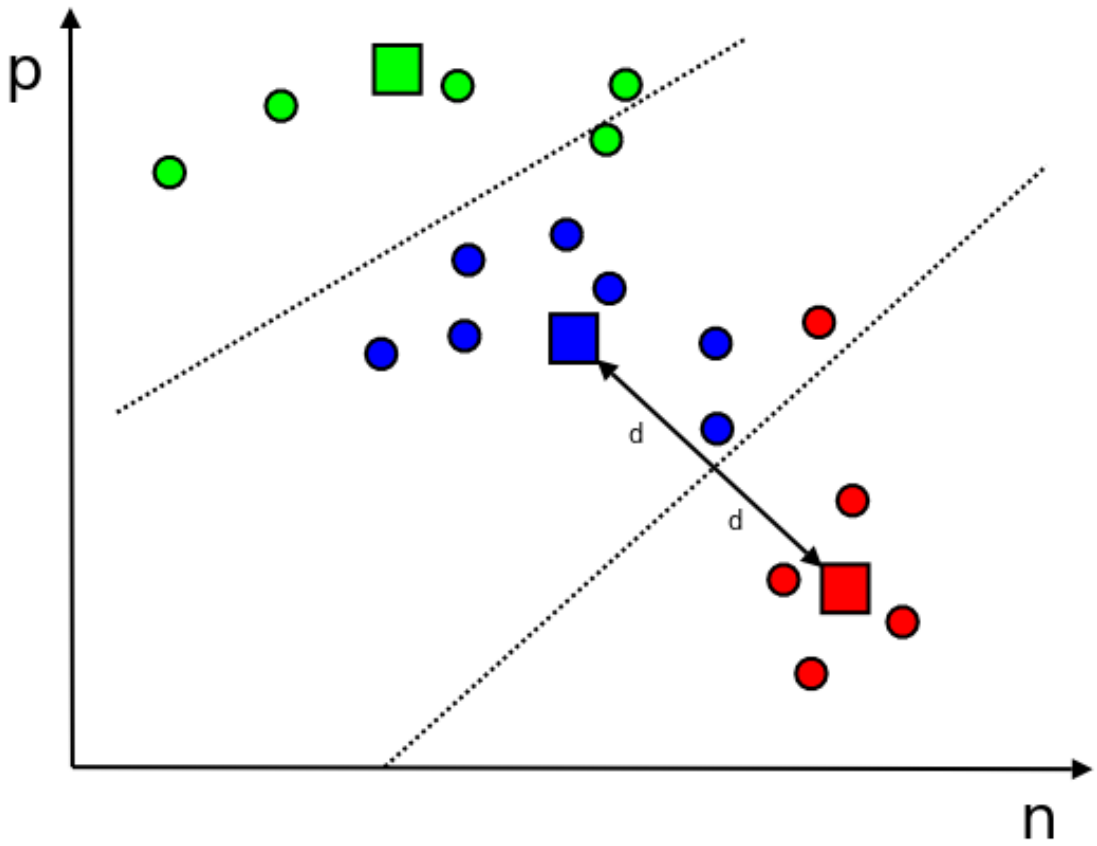


Klasifikace dle etalonů

- Etalony jsou jednoduchým nepravděpodobnostním **modelem** dat.
- Při klasifikaci nových dat již nepoužíváme stará data, ale model.

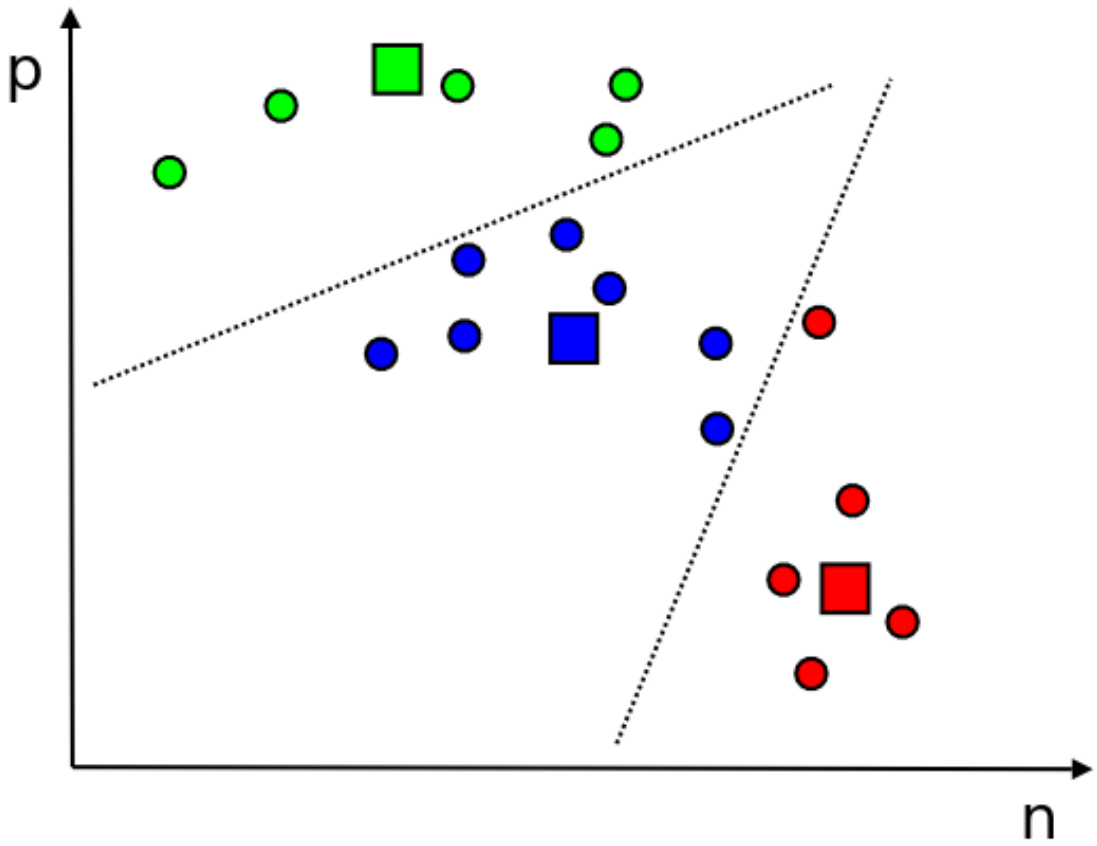


Lineární separace



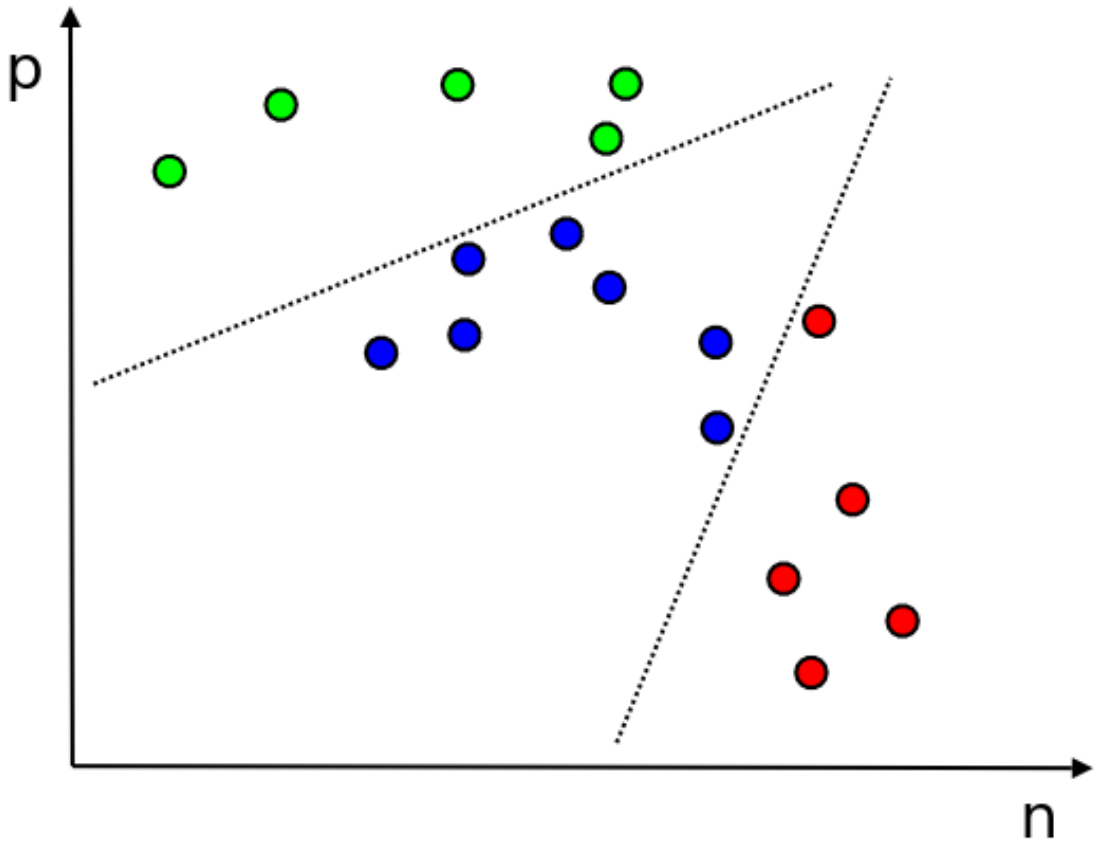
Etalony: lineární separace, chyby na trénovací množině

Lineární separace



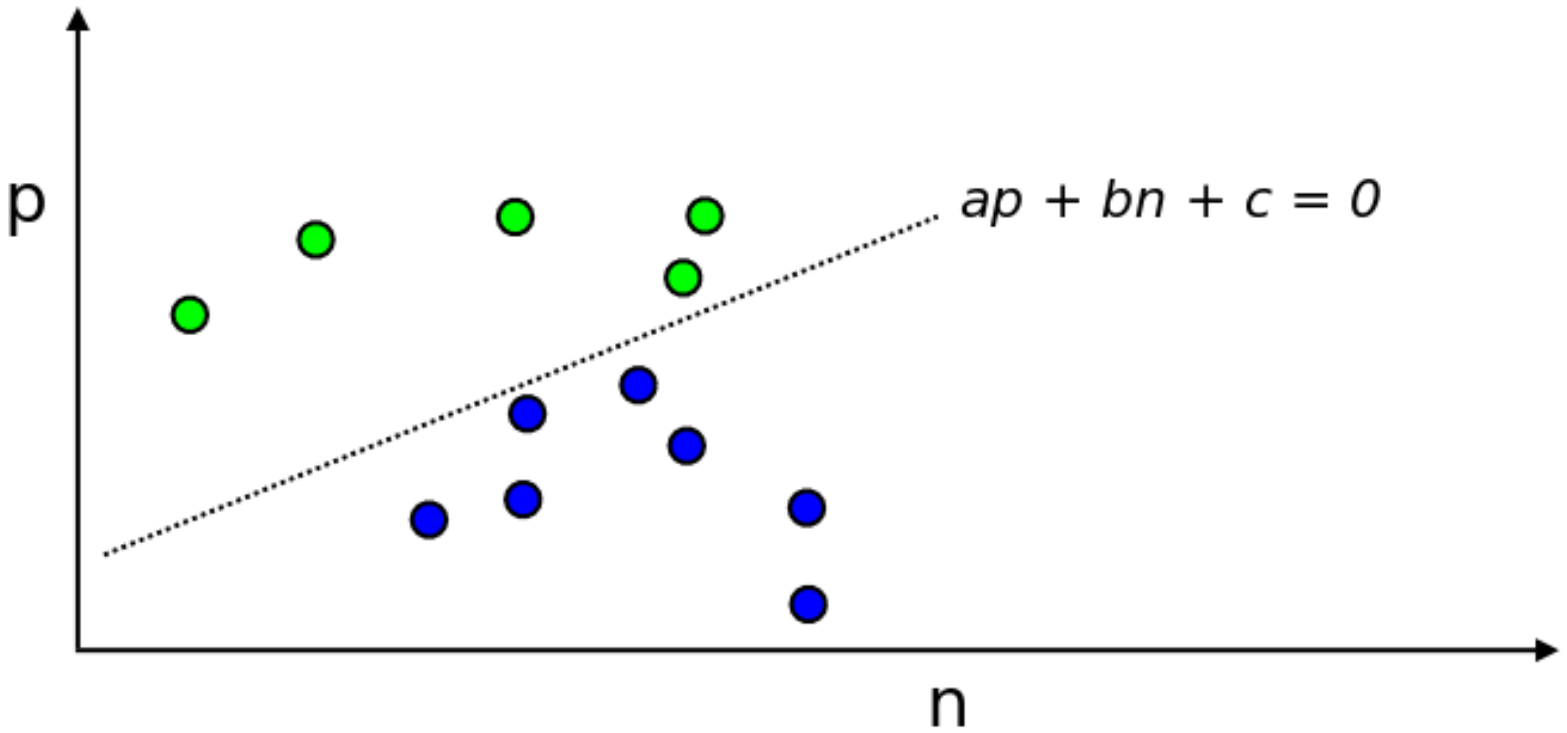
Šlo by to bez chyb, třídy jsou *lineárně separabilní*.

Lineární separace



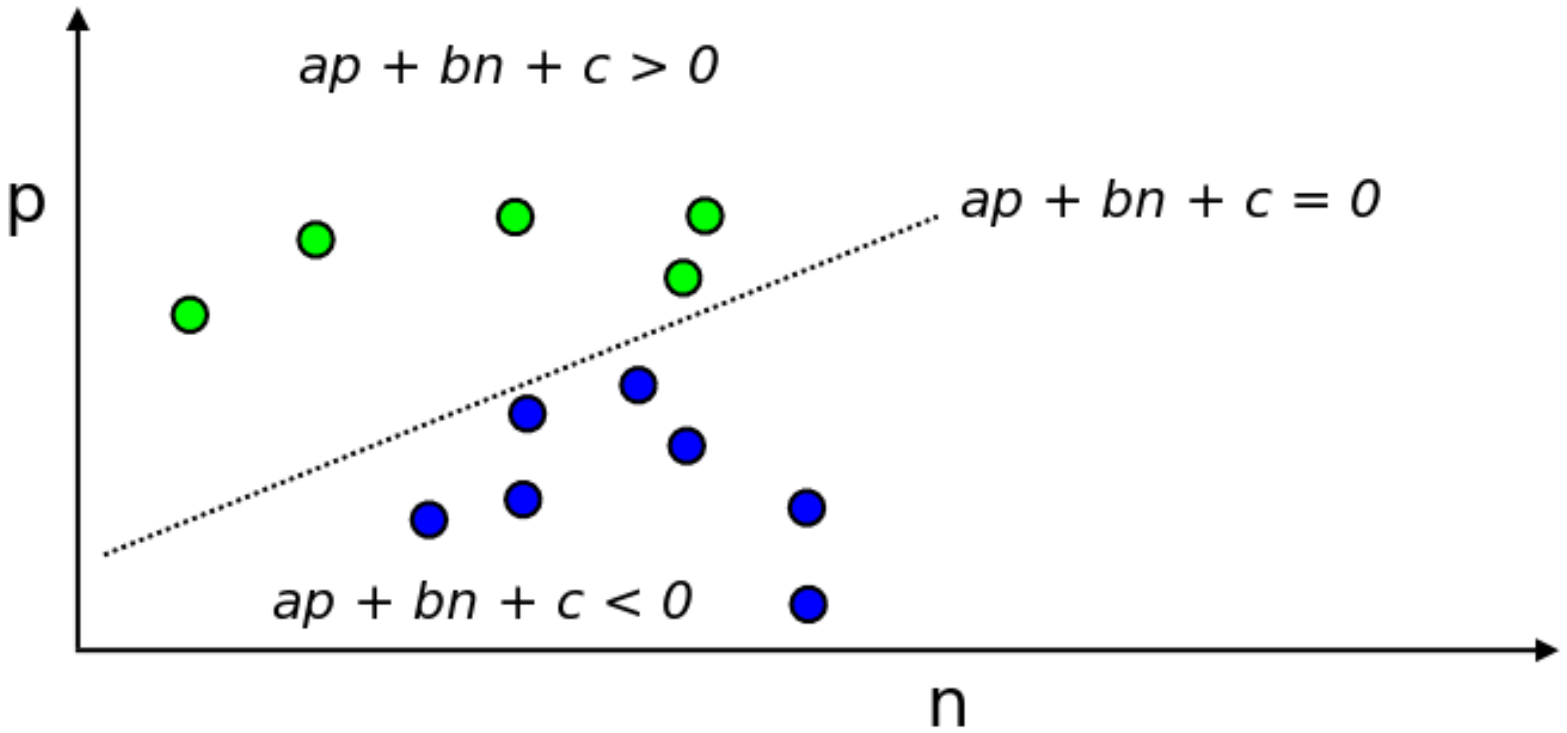
Místo etalonů hledejme modely přímo ve formě separačních přímek.
Příznaky musí být reálné veličiny (až do konce přednášky)
Tedy např. p je příjem v Kč.

Lineární separace dvou tříd



Hledáme parametry a, b, c rovnice přímky.

Lineární separace dvou tříd



Hledáme parametry a, b, c rovnice přímky.

Skalární součin

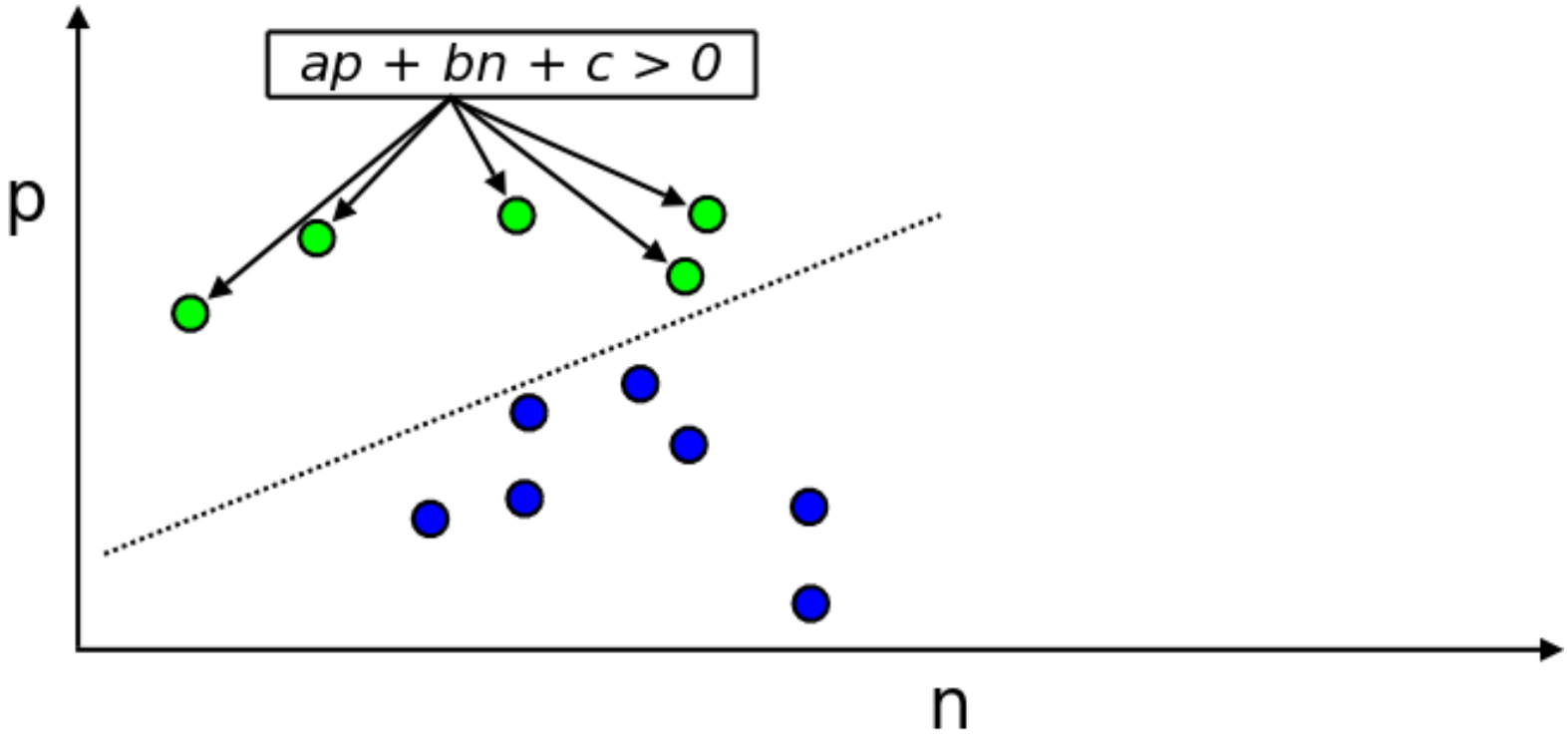
Nechť

$$\vec{\alpha} = (a, b, c, \ldots)$$
$$\vec{\beta} = (m, n, o, \ldots)$$

Definujeme:

$$\vec{\alpha} * \vec{\beta} = a \cdot m + b \cdot n + c \cdot o + \ldots$$

Lineární separace dvou tříd

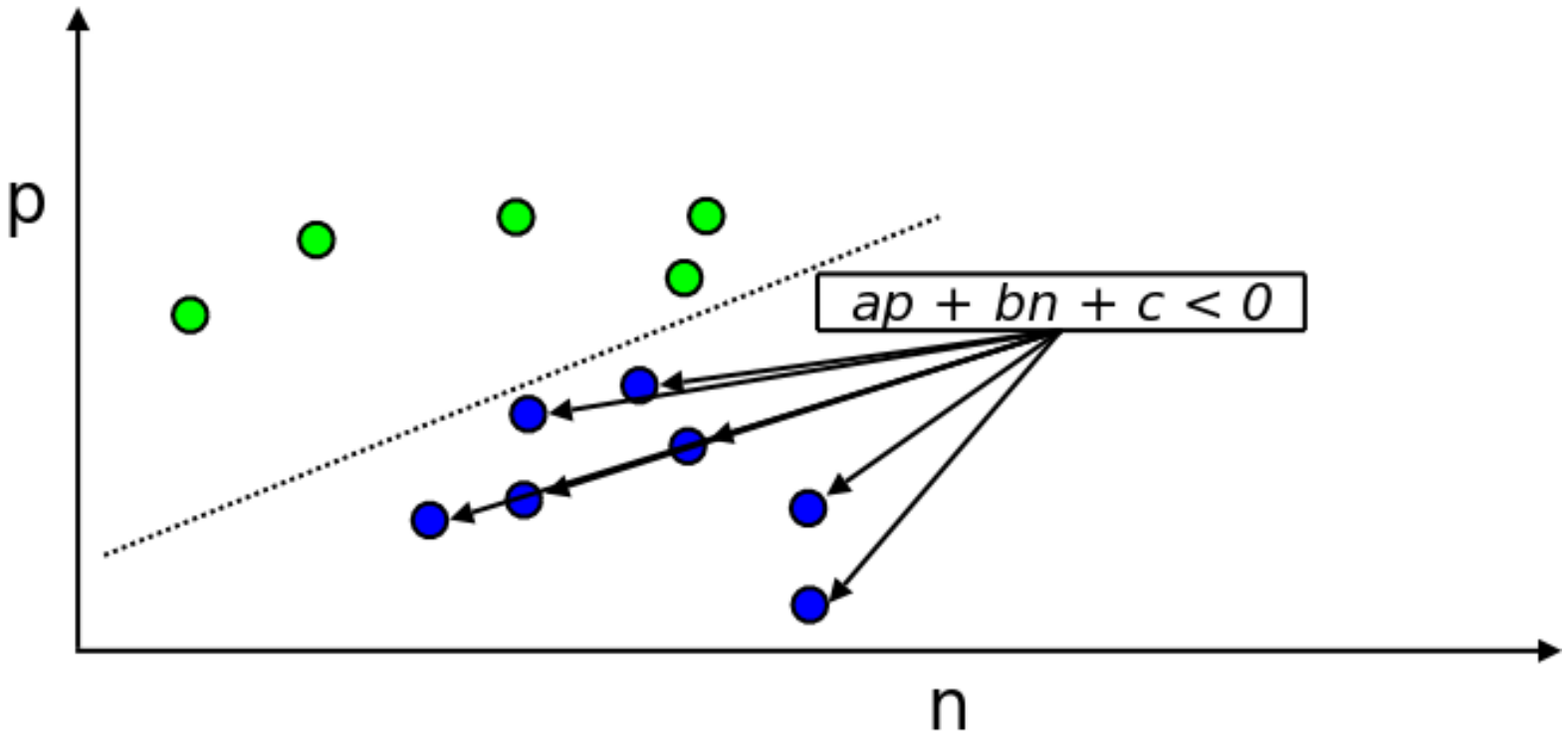


Pro každý příklad klienta třídy splácí s příznaky p_i, n_i:

$$\vec{w} * \vec{x}_i > 0$$

kde $\vec{w} = (a, b, c)$, $\vec{x}_i = (p_i, n_i, 1)$, $i = 1 \ldots 5$.

Lineární separace dvou tříd

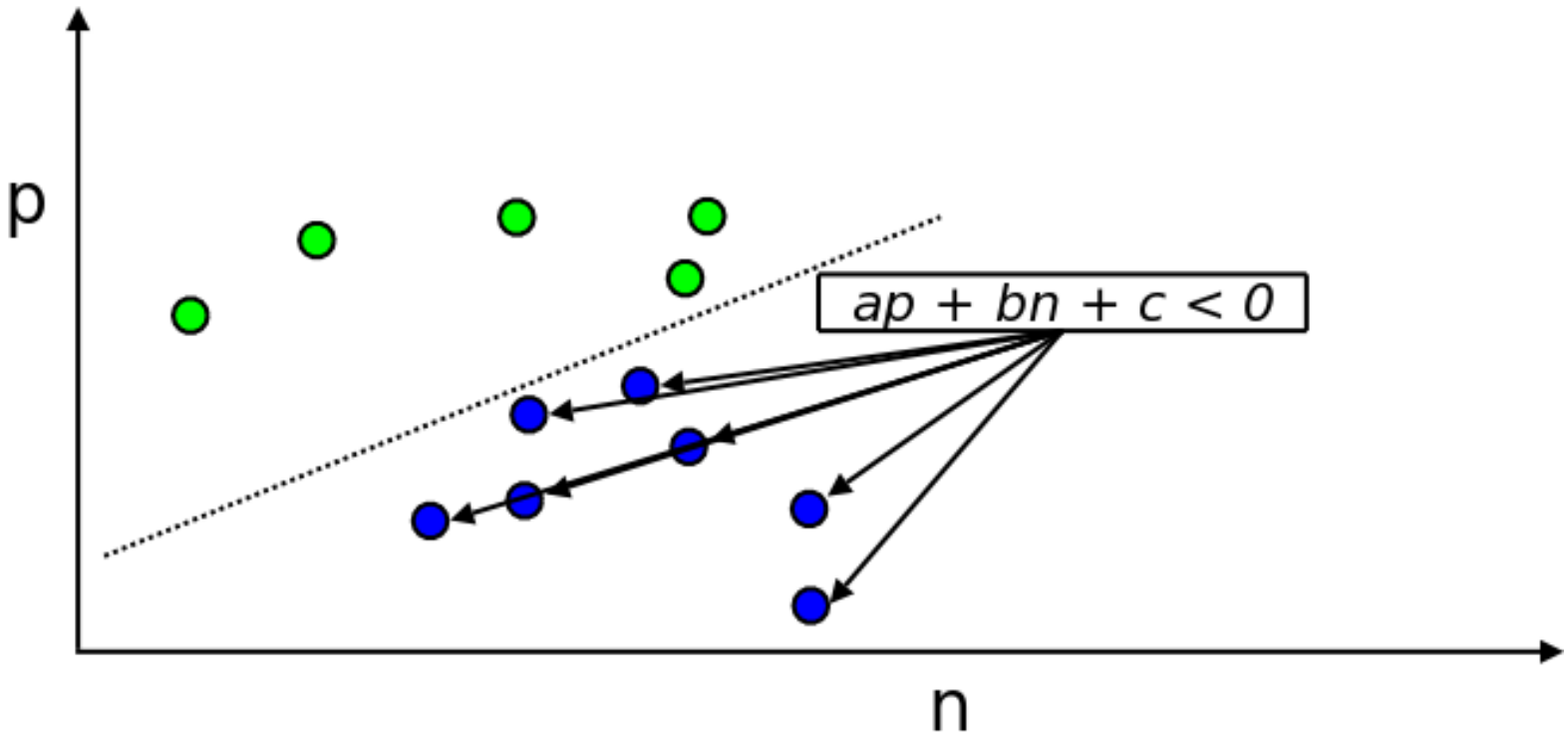


Pro každý příklad klienta třídy problémy s příznaky p_i, n_i:

$$\vec{w} * \vec{x}_i < 0$$

kde $\vec{w} = (a, b, c)$, $\vec{x}_i = (p_i, n_i, 1)$, $i = 1 \ldots 7$.

Lineární separace dvou tříd

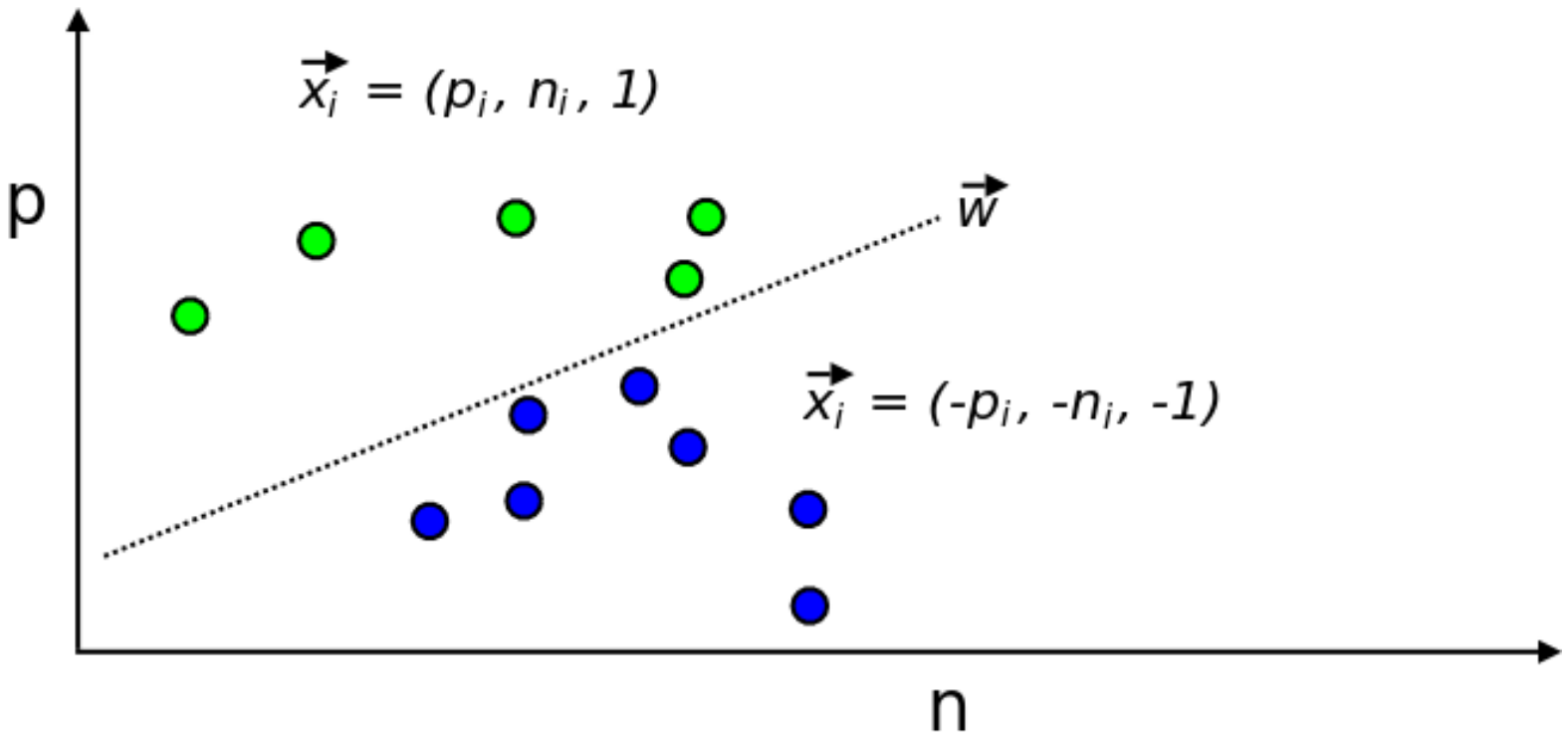


Pro každý příklad klienta třídy problémy s příznaky p_i, n_i:

$$\vec{w} * \vec{x}_i > 0$$

kde $\vec{w} = (a, b, c)$, $\vec{x}_i = (-p_i, -n_i, -1)$, $i = 1 \ldots 7$.

Lineární separace dvou tříd

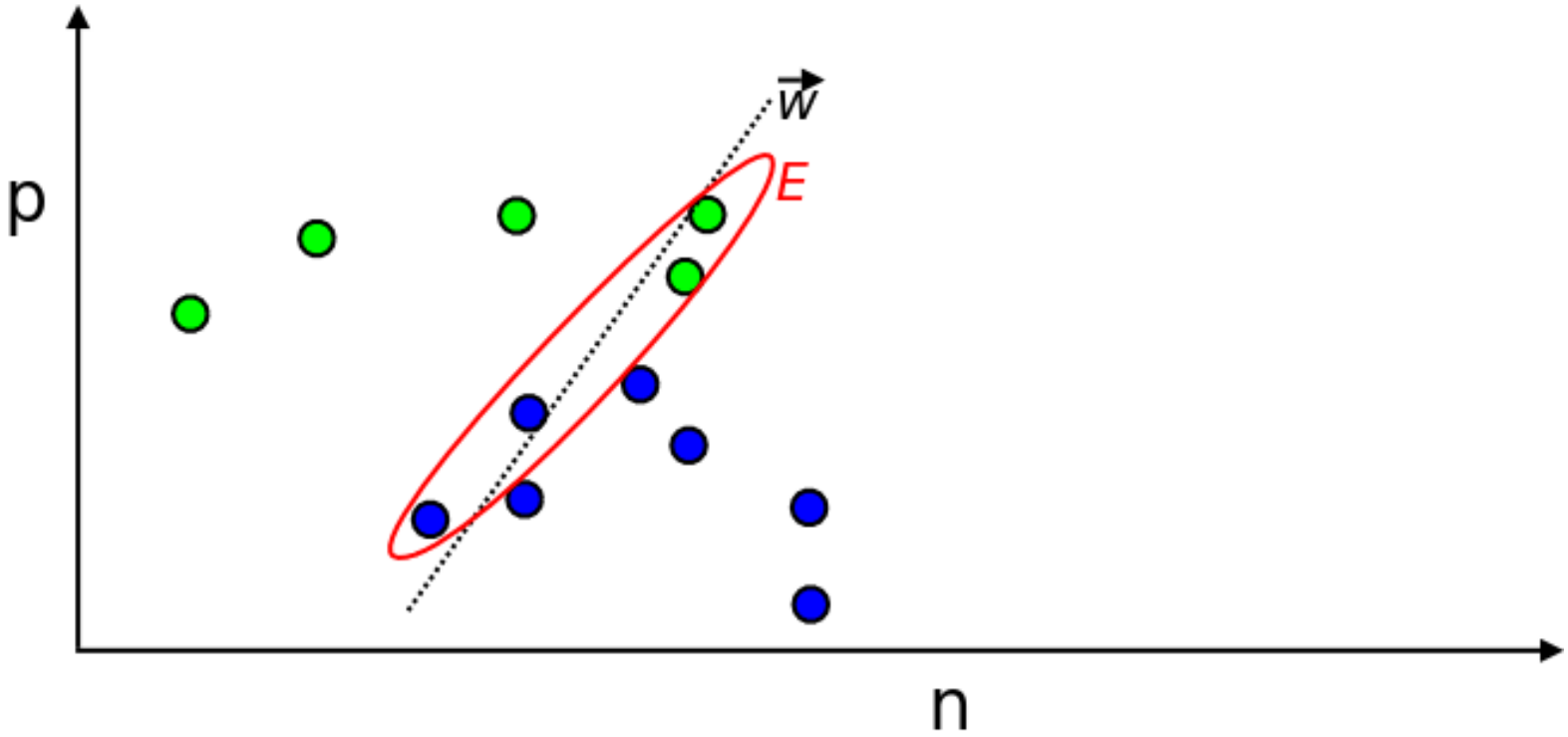


Hledáme řešení soustavy

$$\vec{w} * \vec{x_i} > 0$$

$$i = 1 \ldots 12$$

Chybová funkce

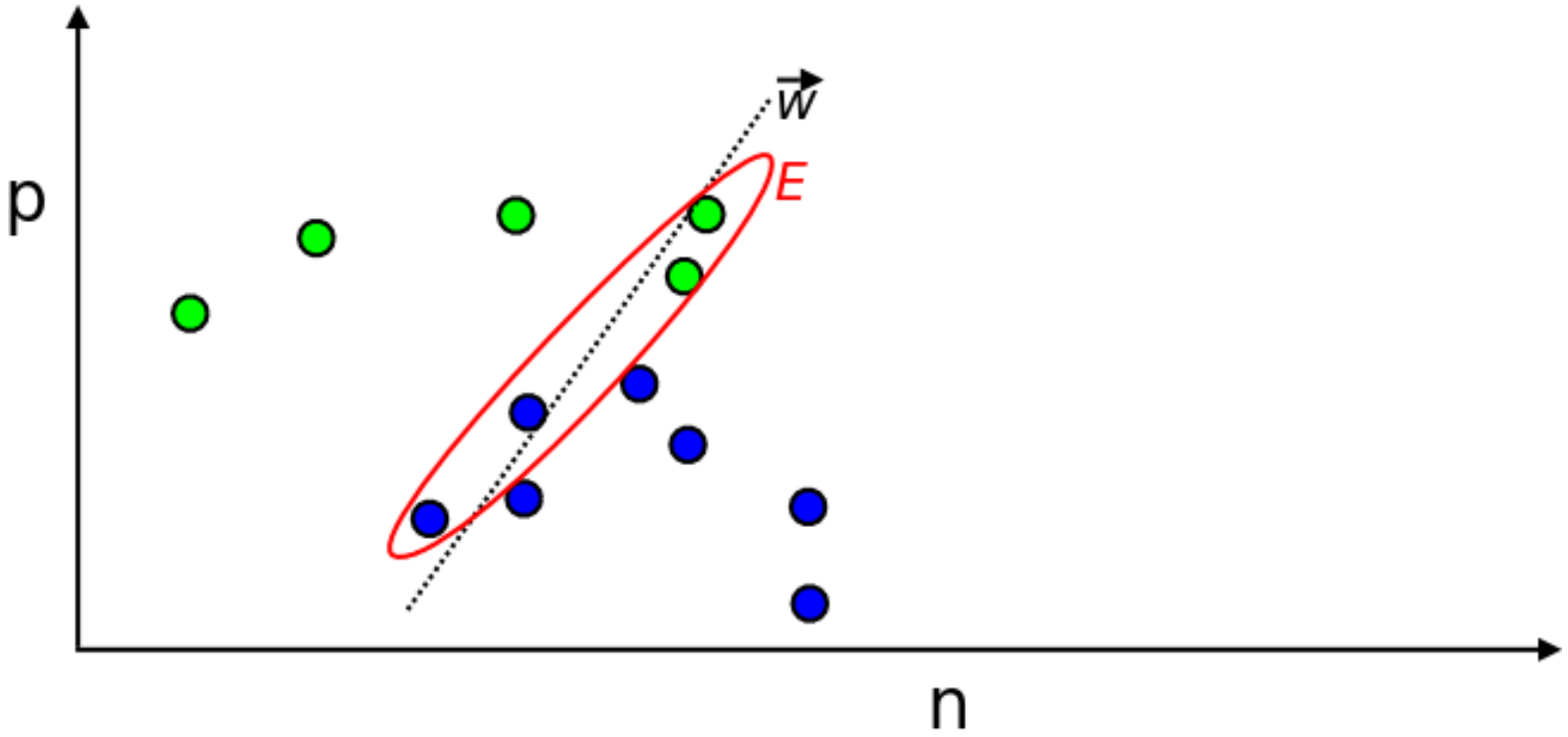


Nechť E je množina příkladů nesprávně klasifikovaných přímkou $\vec{w}$. Nechť

$$e(\vec{w}) = \sum_{\vec{x}_i \in E} -\vec{w} * \vec{x}_i$$

$e(\vec{w}) \geq 0$ vždy a $e(\vec{w}) = 0$, jsou-li všechny příklady klasifikovány správně.

Chybová funkce

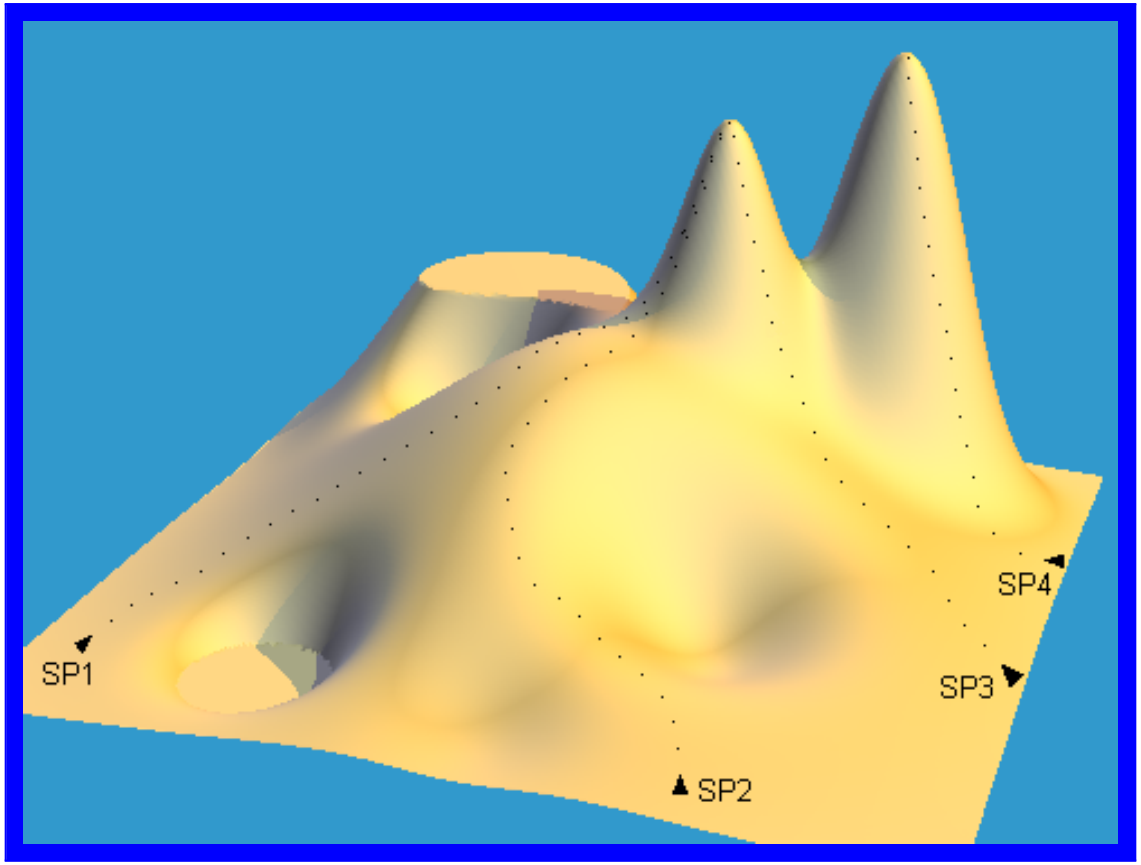


Hledáme tedy minimum funkce

$$e(\vec{w}) = \sum_{\vec{x}_i \in E} -\vec{w} * \vec{x}_i$$

mezi všemi $\vec{w} \in R^3$

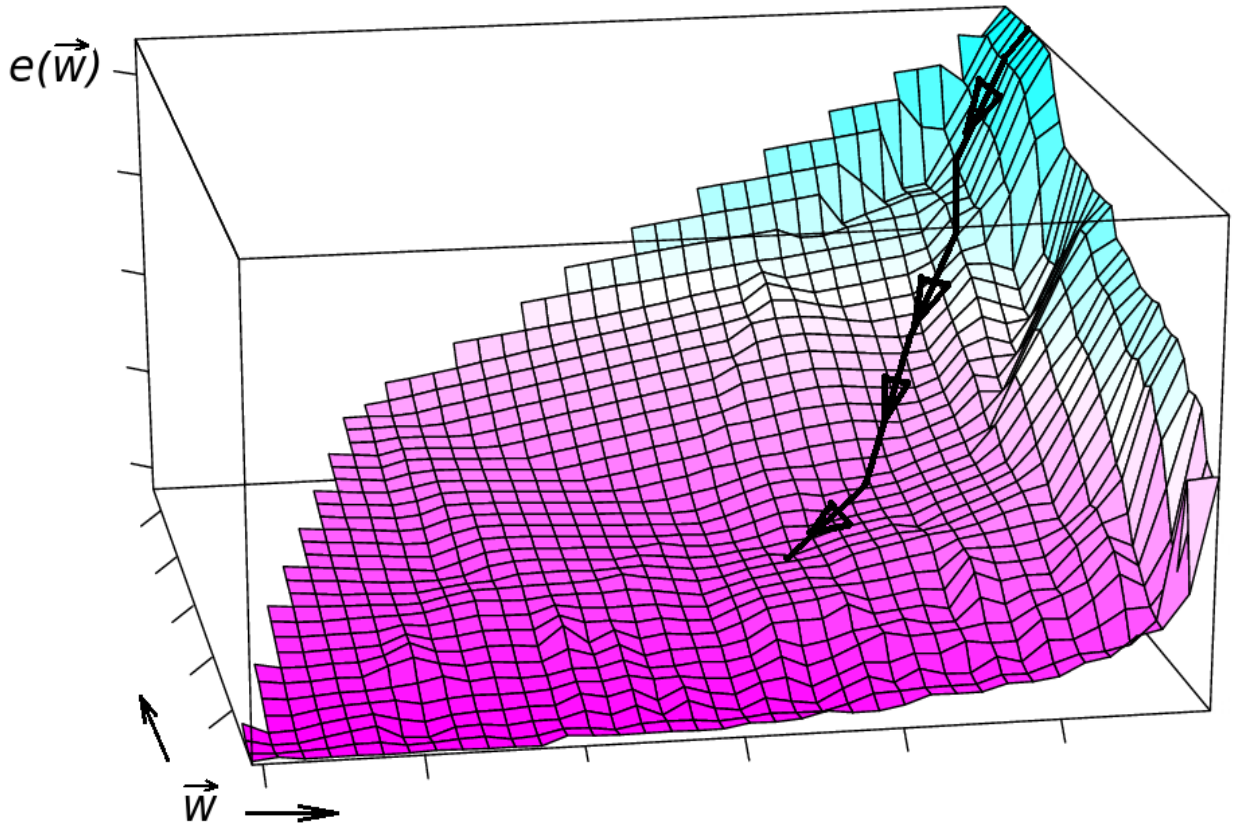
Gradient



Směr největšího vzrůstu funkce $f(\vec{\alpha})$ proměnné $\vec{\alpha} = (\alpha_1, \alpha_2, \dots)$

$$\nabla f(\vec{\alpha}) = \left(\frac{\partial f(\vec{\alpha})}{\partial \alpha_1}, \frac{\partial f(\vec{\alpha})}{\partial \alpha_2}, \dots \right)$$

Gradientní hledání minima $e(\vec{w})$



V každém kroku $k = 1, 2, \ldots$:

$$\vec{w}(k+1) = \vec{w}(k) - \eta \cdot \nabla e(\vec{w}(k))$$

$\eta \in R$ je velikost kroku

Gradientní hledání minima $e(\vec{w})$

$$\nabla e(\vec{w}) = \nabla \sum_{x_i \in E} -\vec{w} * \vec{x_i} = \sum_{x_i \in E} -\vec{x_i}$$

Gradientní hledání minima $e(\vec{w})$

$$\nabla e(\vec{w}) = \nabla \sum_{x_i \in E} -\vec{w} * \vec{x_i} = \sum_{x_i \in E} -\vec{x_i}$$

V každém kroku $k = 1, 2 \ldots$ tedy:

$$\vec{w}(k+1) = \vec{w}(k) + \eta \cdot \sum_{x_i \in E} \vec{x_i}$$

Gradientní hledání minima $e(\vec{w})$

$$\nabla e(\vec{w}) = \nabla \sum_{x_i \in E} -\vec{w} * \vec{x_i} = \sum_{x_i \in E} -\vec{x_i}$$

V každém kroku $k = 1, 2 \ldots$ tedy:

$$\vec{w}(k+1) = \vec{w}(k) + \eta \cdot \sum_{x_i \in E} \vec{x_i}$$

Varianta s proměnnou velikostí kroku:

$$\vec{w}(k+1) = \vec{w}(k) + \eta(k) \cdot \sum_{x_i \in E} \vec{x_i}$$

Gradientní hledání minima $e(\vec{w})$

$$\nabla e(\vec{w}) = \nabla \sum_{x_i \in E} -\vec{w} * \vec{x_i} = \sum_{x_i \in E} -\vec{x_i}$$

V každém kroku $k = 1, 2 \ldots$ tedy:

$$\vec{w}(k+1) = \vec{w}(k) + \eta \cdot \sum_{x_i \in E} \vec{x_i}$$

Varianta s proměnnou velikostí kroku:

$$\vec{w}(k+1) = \vec{w}(k) + \eta(k) \cdot \sum_{x_i \in E} \vec{x_i}$$

Aby $\lim_{k \to \infty} \vec{w}(k)$ konvergovala, určíme $\eta(k)$ tak, aby

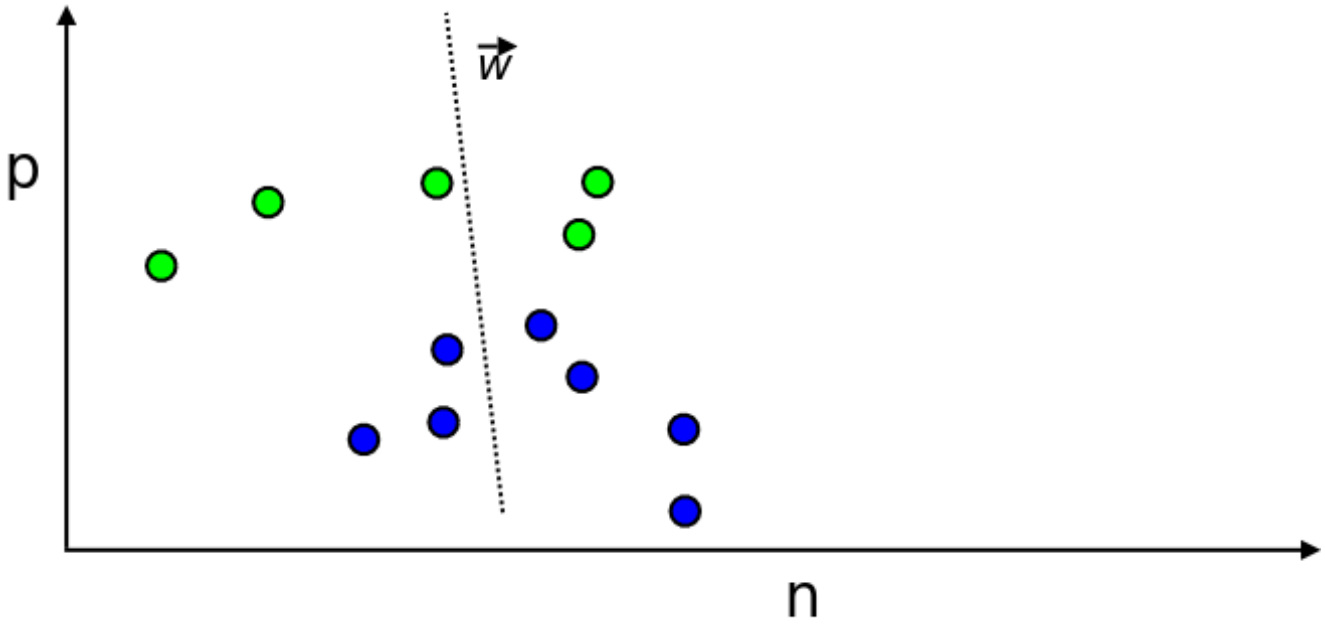
$$\lim_{k \to \infty} \eta(k) = 0$$

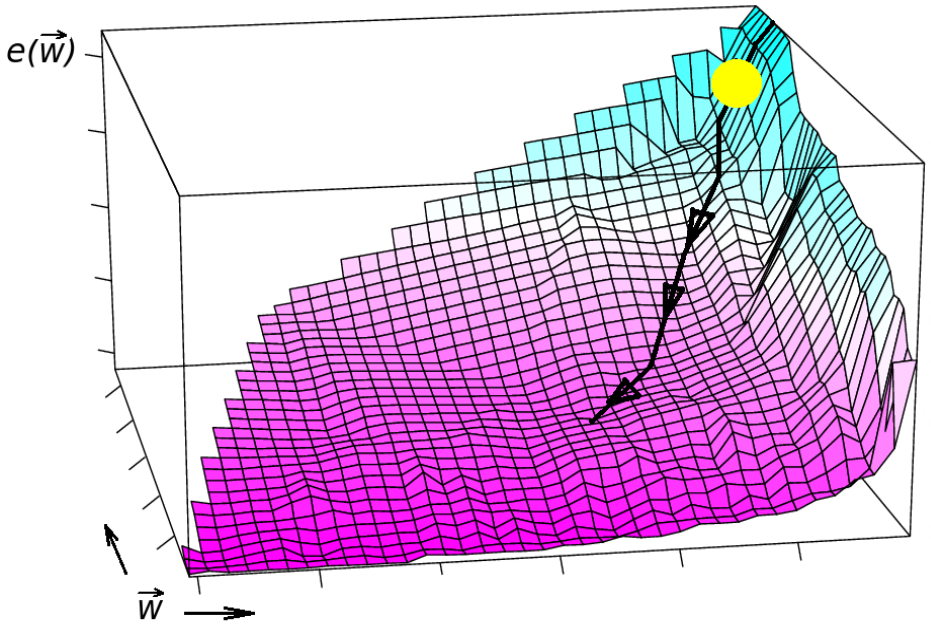
Gradientní (perceptronový) algoritmus

Perceptronový algoritmus

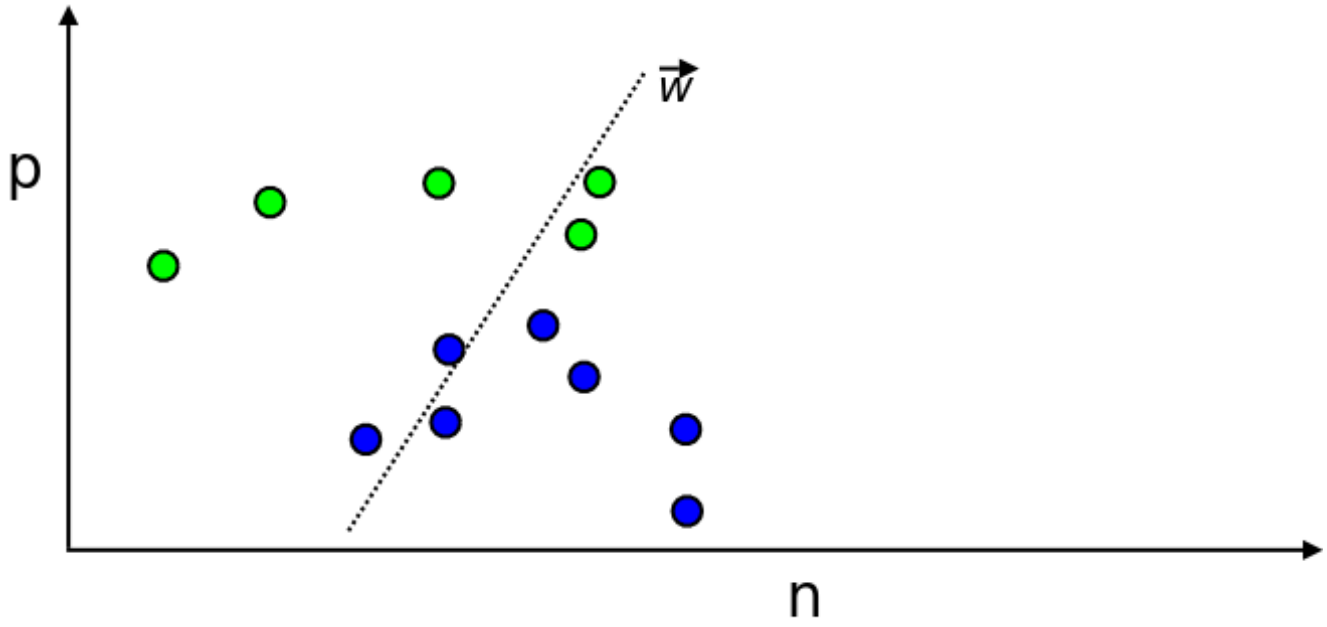
Input: $\vec{w}$, $\eta(.)$, θ
Output: $\vec{w}$
repeat
  $k \leftarrow k + 1$
  $\vec{w} \leftarrow \vec{w} + \eta(k) \sum_{\vec{x}_i \in E} \vec{x}_i$
until $|\eta(k) \sum_{\vec{x}_i \in E} \vec{x}_i| < \theta$;
return $\vec{w}$

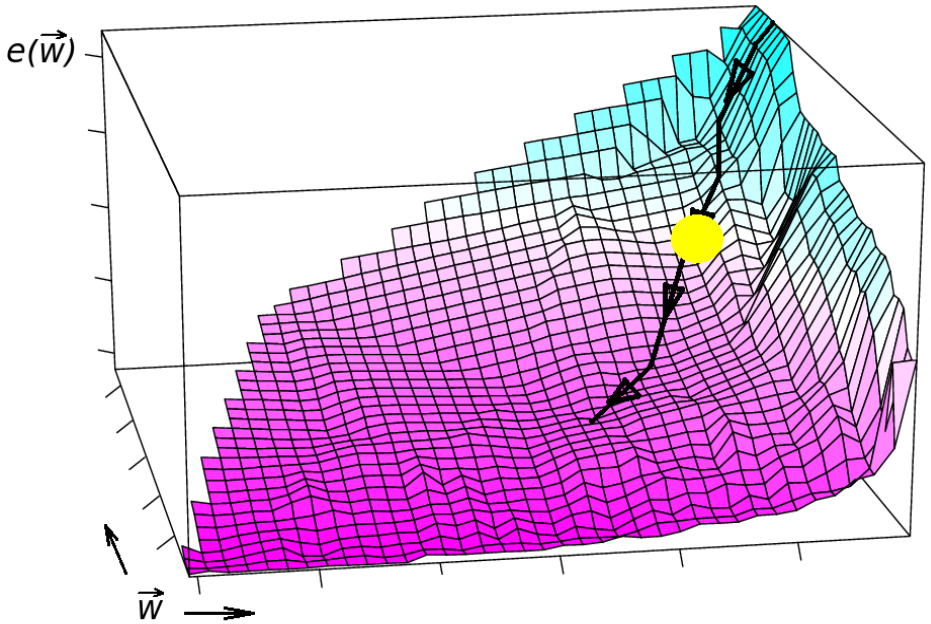
Gradientní (perceptronový) algoritmus



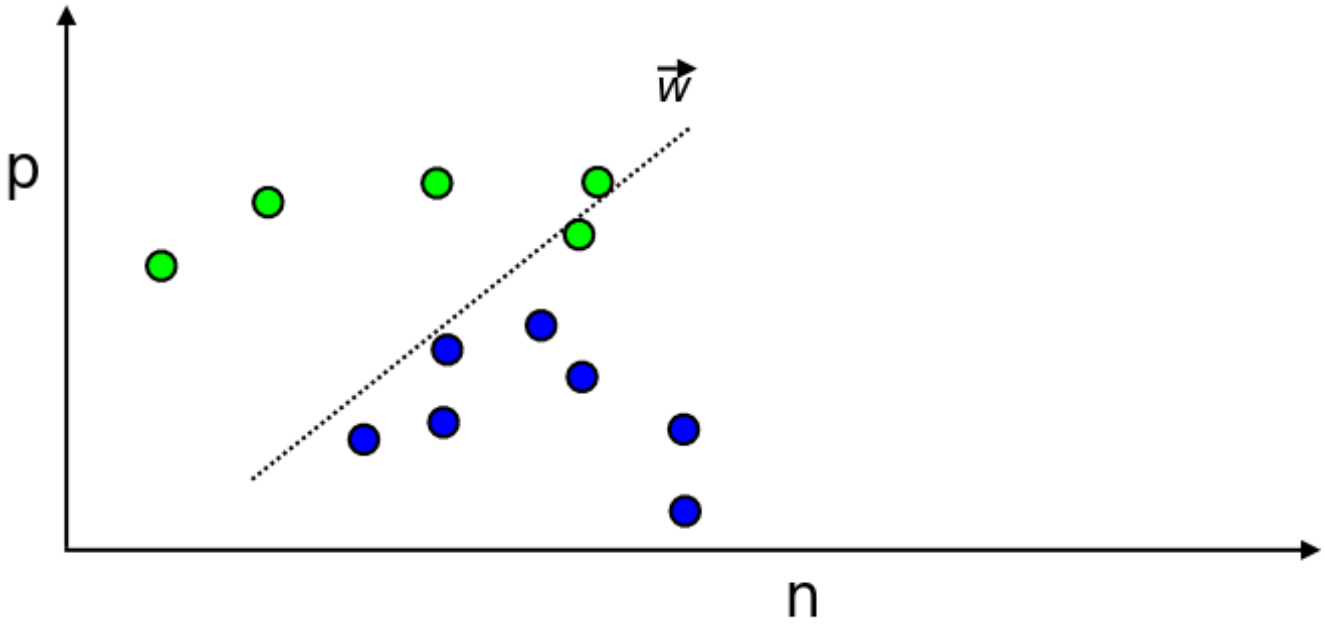


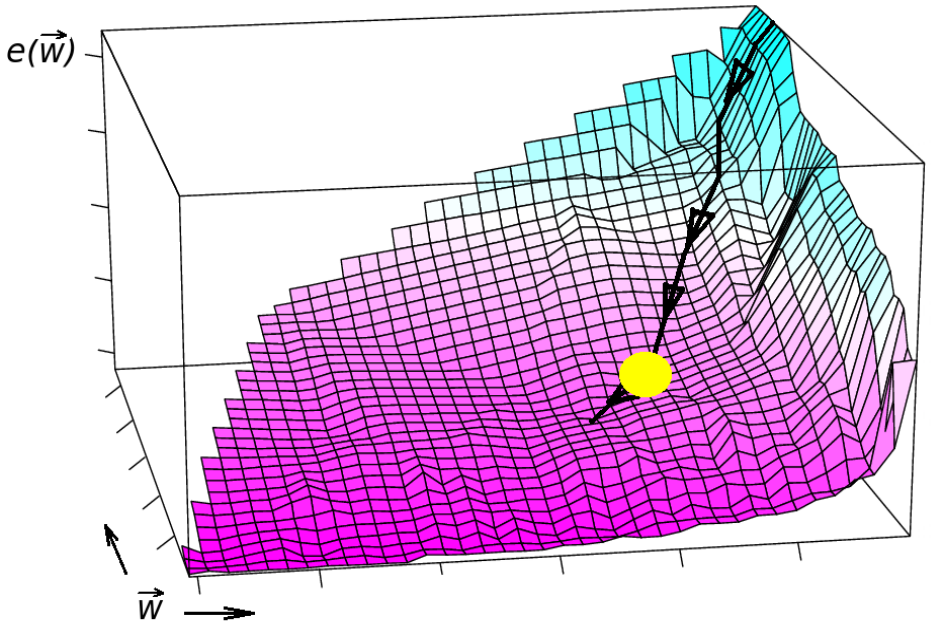
Gradientní (perceptronový) algoritmus



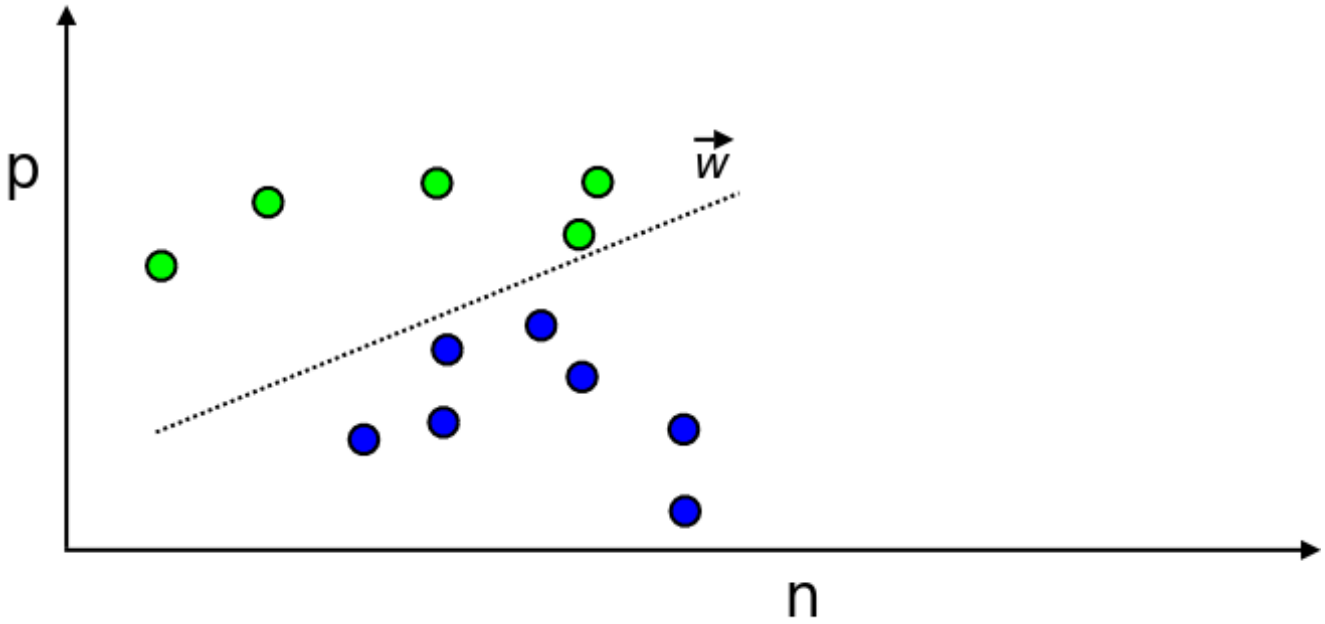


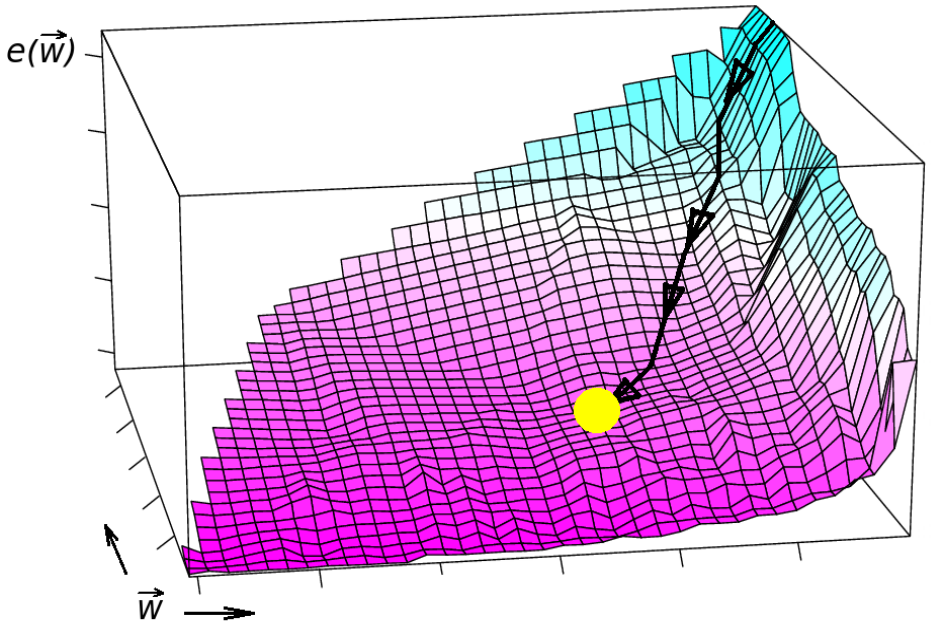
Gradientní (perceptronový) algoritmus





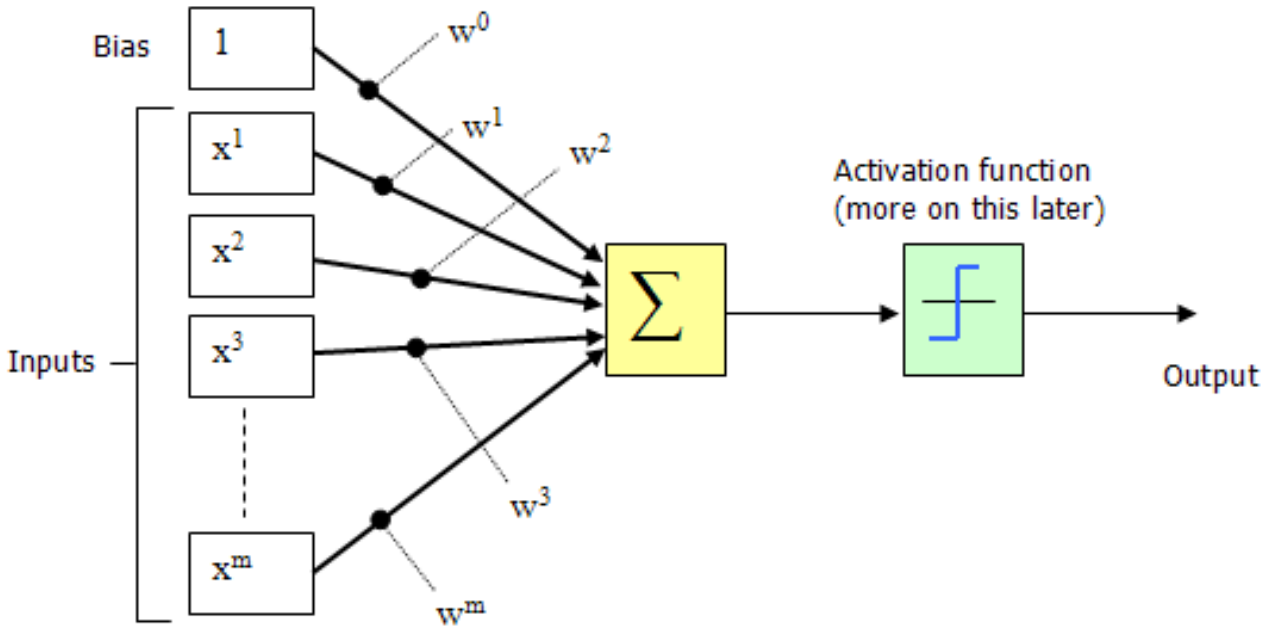
Gradientní (perceptronový) algoritmus





Perceptron

Rovnice v grafické představě



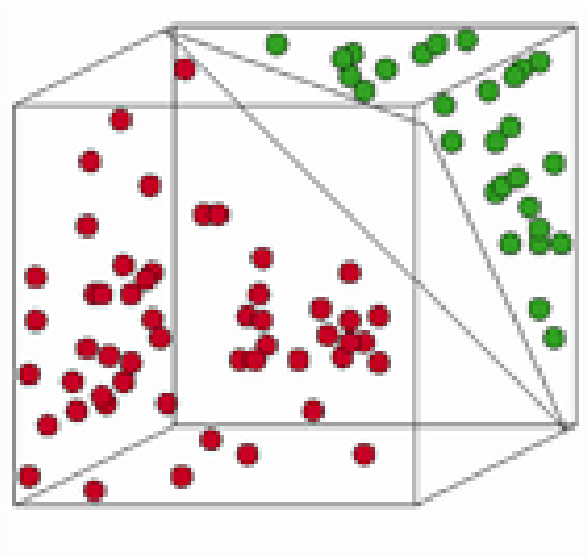
Základ modelu umělého neuronu

Vyšší dimenze

- Příklady dosud dvourozměrné (příznaky p, n)
- Ve vyšších dimenzích stejné principy
- Tři příznaky: místo přímky hledáme

Vyšší dimenze

- Příklady dosud dvourozměrné (příznaky p, n)
- Ve vyšších dimenzích stejné principy
- Tři příznaky: místo přímky hledáme separační rovinu

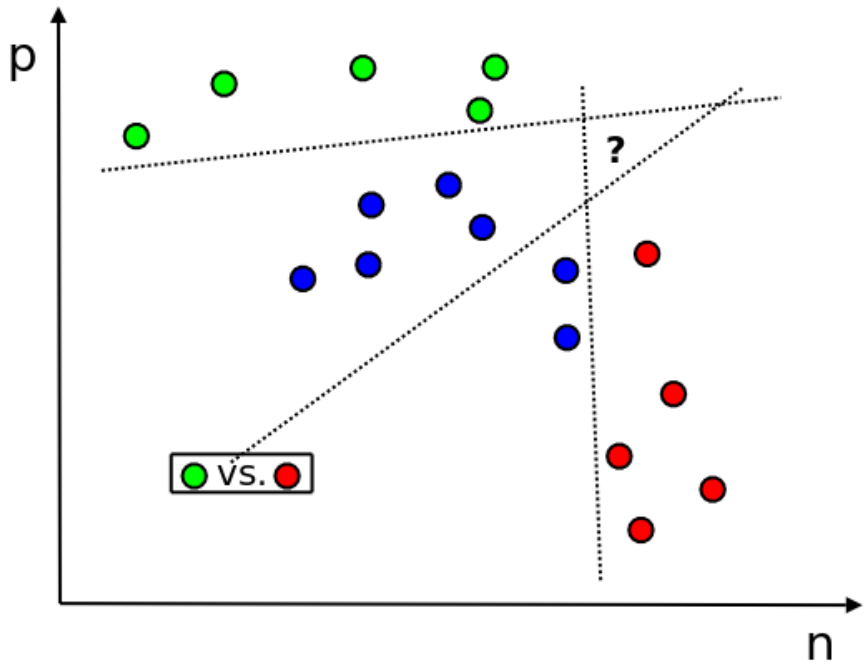


- Více příznaků: hledáme tzv. nadrovinu (hyperrovinu)

Více tříd

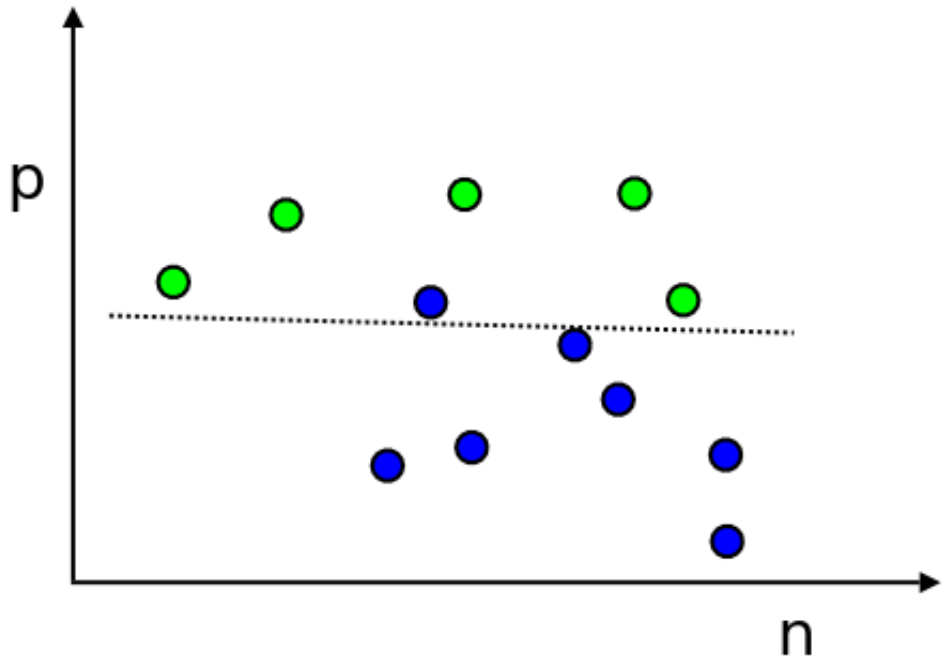
- Dosud: klasifikace mezi dvěma třídami. Možnosti pro 3 a více tříd?

- *„1 proti všem":* T nadrovin pro T tříd, každá odděluje jednu od ostatních.
  - Problém: jak rozhodnout při klasifikaci, pokud „vyhraje" více než 1 třída?
- *Dvojice:* $T(T-1)/2$ nadrovin pro T tříd, každá odděluje jeden pár tříd.
  - Problém: vznikají oblasti nejjednoznačnosti $\rightarrow$



- Řešení pomocí *diskriminačních funkcí* mimo rozsah přednášky.

Lineárně neseparabilní třídy



Možnosti

- Spokojit se s nenulovou trénovací chybou
- Separovat nelineárně

Rozšíření báze

Převádí **nelineární** separaci na **lineární** separaci ve **vyšším rozměru**.

- K množině příznaků (p, n) přidáme součiny do s-tého stupně.
- Pro $s = 2$ nová množina (p, n, o, q, r), kde

$$o = p \cdot p = p^2$$
$$q = p \cdot n$$
$$r = n \cdot n = n^2$$

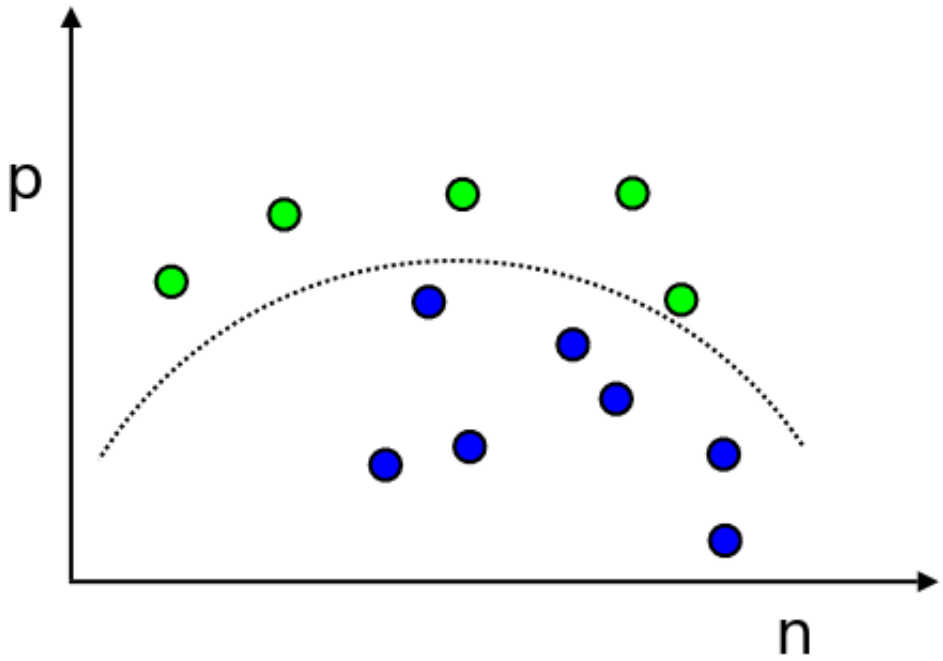
- Např. perceptronem obdržíme separující nadrovinu v 5D:

$$a \cdot p + b \cdot n + c \cdot o + d \cdot q + e \cdot r + f$$

- tj. separující polynom stupně 2 zpět ve 2D:

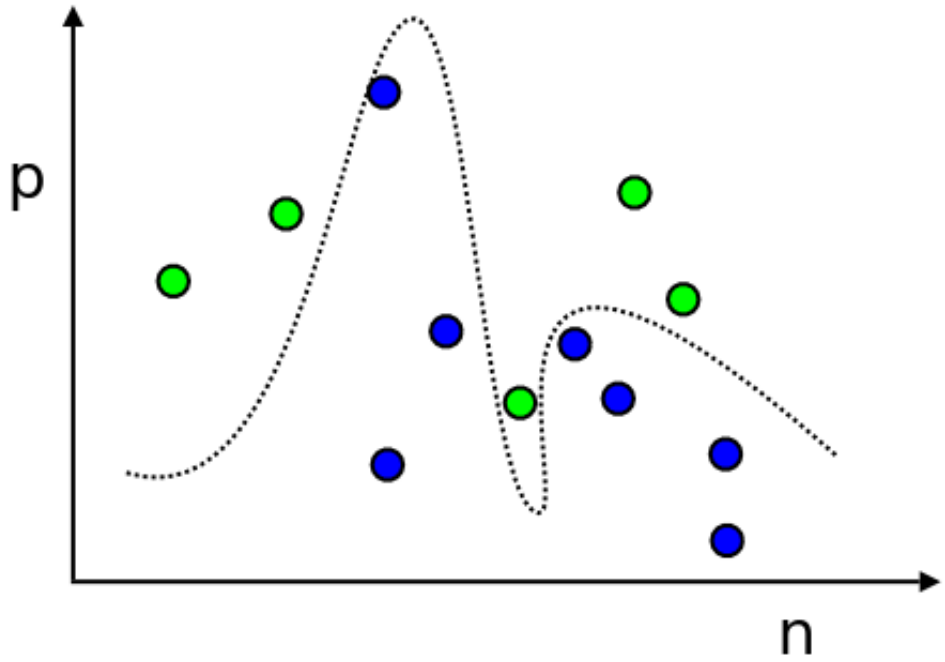
$$c \cdot p^2 + e \cdot n^2 + d \cdot p \cdot n + a \cdot p + b \cdot n + f$$

Kvadratická separace



- Pokud by trénovací chyba nekonvergovala k nule ani s kvadratickou separací, můžeme dále rozšířovat bázi na $s = 3, 4, \ldots$.

Separace polynomem vyššího řádu



- V čem je nebezpečí zvyšování s?

Trénovací vs. skutečná chyba

- Typický průběh chyby pro vzrůstající stupeň separačního polynomu
- Přeučení
- Srovnejte s podobnou analogickou závislostí u klasifikace dle sousedů