Strojové učení a rozpoznávání

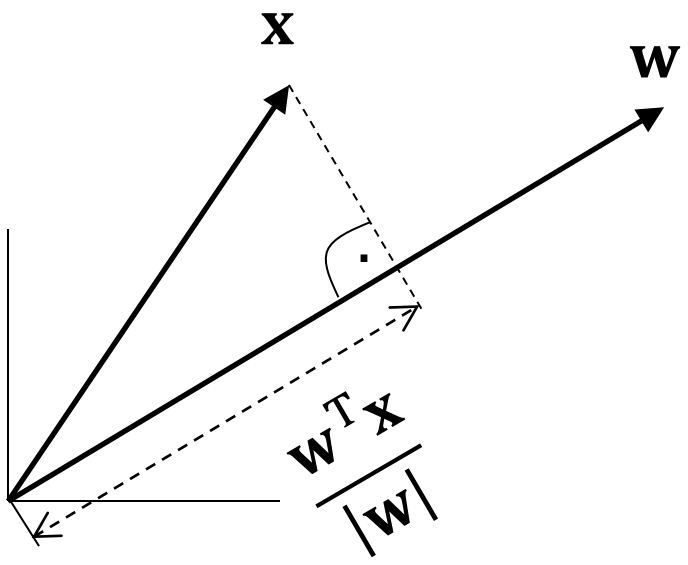
Lineární klasifikátory

Lukáš Burget

Opakování - Skalární součin

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} = \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = w_1 x_1 + w_2 x_2$$



Lineární klasifikátor

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

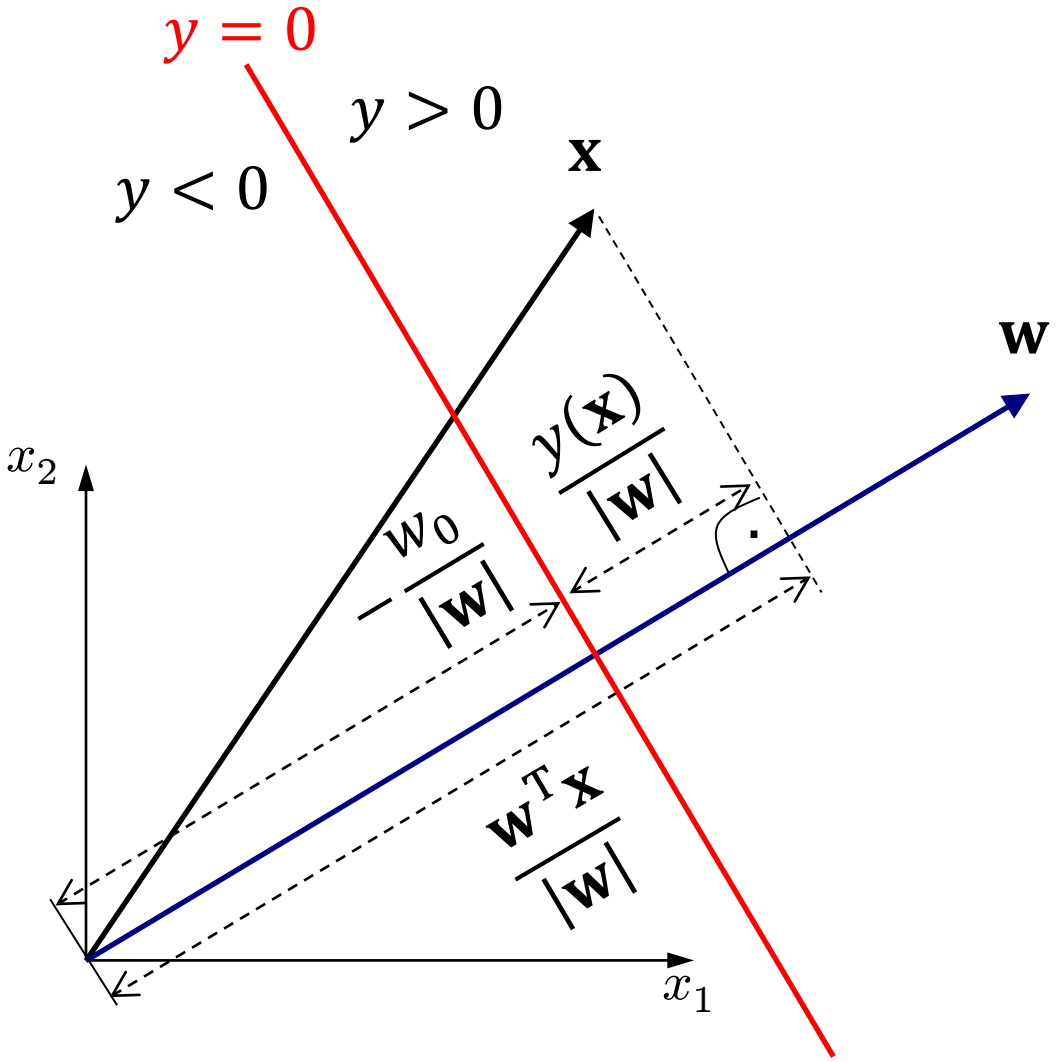
Vyber třídu C_1 pokud $y(\mathbf{x}) > 0$ a jinak vyber třídu C_2

Zobecněný lineární klasifikátor

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right)$$

kde f se nazývá aktivační funkce

Lineární klasifikátor



$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$$

$$\frac{\mathbf{w}^T\mathbf{x}}{|\mathbf{w}|} = -\frac{w_0}{|\mathbf{w}|} + \frac{y(\mathbf{x})}{|\mathbf{w}|}$$

Perceptron

- Jednoduchý lineární klasifikátor s aktivační funkcí:

$$f(a) = \begin{cases} +1, & a \geqslant 0 \\ -1, & a < 0 \end{cases}$$

- Samotná aktivační funkce v tomto případě nic nezmění – rozhodování na základě y(**x**) > 0 by vedlo ke stejnému výsledku – ale pro učící se algoritmus bude výhodné definovat si požadovaný výstup jako:

$$t \in \{-1, +1\}$$

- Pro další zjednodušení předpokládejme, ze w_0 je "nultý" koeficient vektoru **w** a odpovídající vstup x_0 je vždy 1. Můžeme tedy psát pouze:

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}\right)$$

Perceptron – učící algoritmus

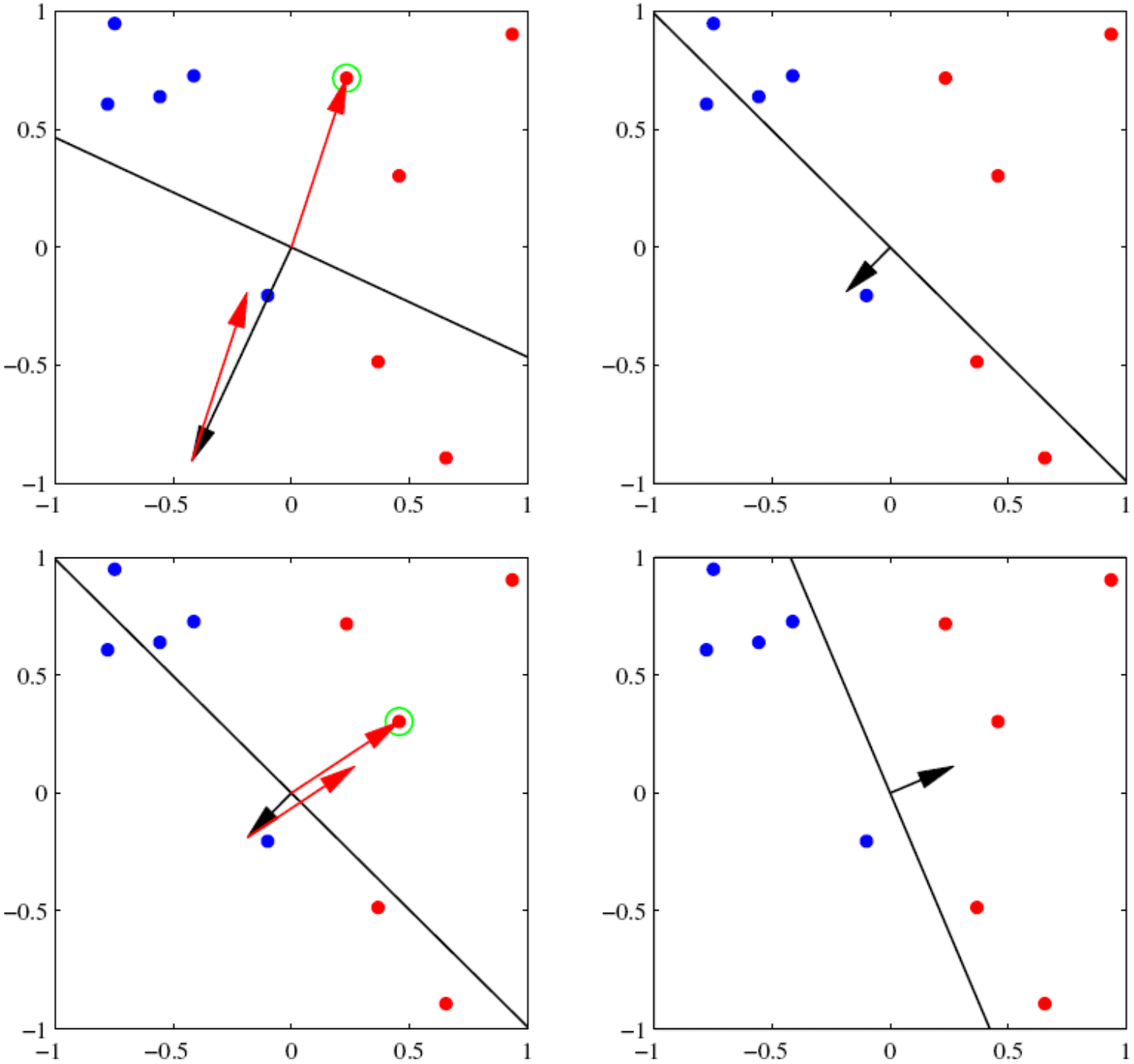
- Cyklicky procházej jednotlivé trénovací vzory $\mathbf{x}_\mathrm{n}$ a vždy když narazíš na špatně klasifikovaný vzor kde

$$y(\mathbf{x}_\mathrm{n}) \neq t_n$$

změň vektor $\mathbf{w}$ takto:

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} + \mathbf{x}_\mathrm{n} t_n$$

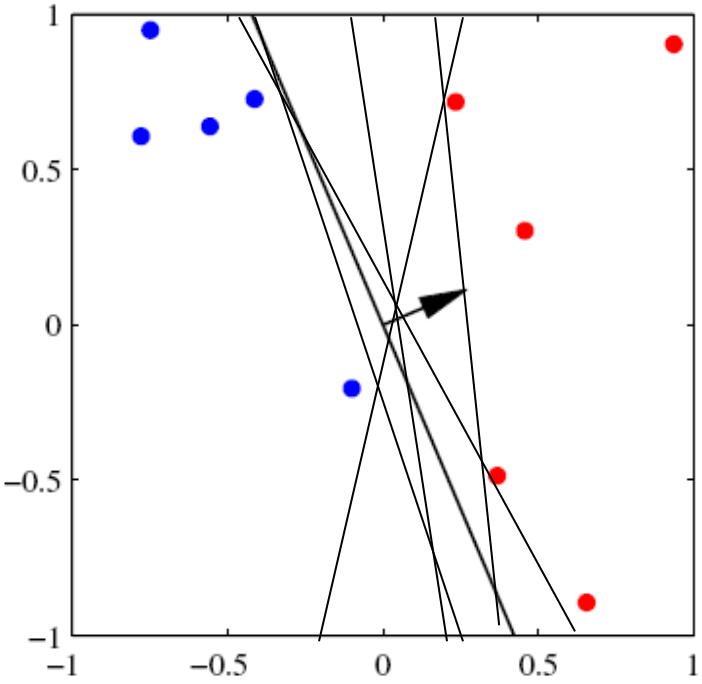
- Lze dokázat, že pokud jsou data lineárně separovatelná, tak, algoritmus vždy nalezne řešení – konverguje.
  V opačném případě, ale nikdy nekonverguje

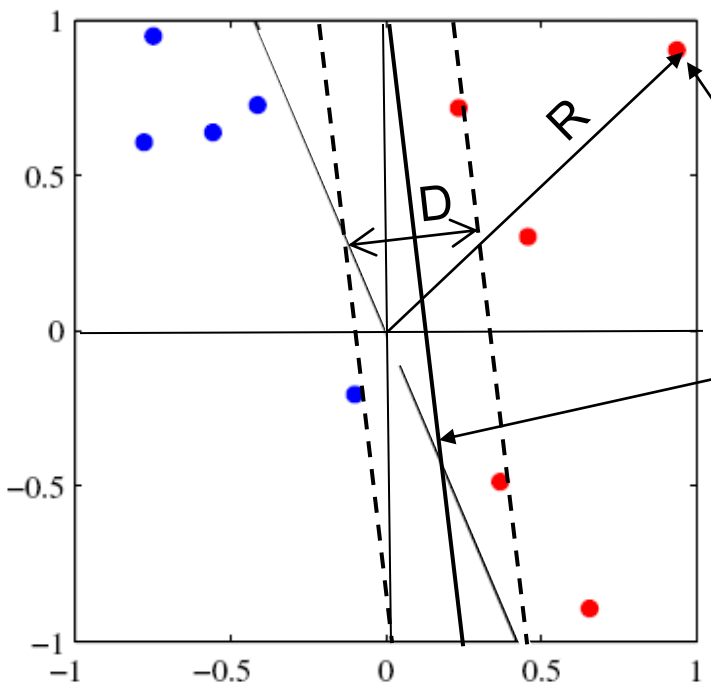

1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1

Perceptron

- Ale které řešení je to správné?
- Řešení, které poskytne učící algoritmus perceptronu záleží na inicializaci – počátečním $\mathbf{w}$

- Algoritmus konverguje v méně než $(R/D)^2$ krocích





Nejvzdálenější bod od počátku

SVM řešení

Opakování - MAP klasifikátor

- Mějme 2 třídy C_1 a C_2
  - Pro daný příznak x vyber třídu C s větší posteriorní pravděpodobností $p(C|\mathbf{x})$
  - Vyber C_1 pouze pokud:

$$P(C_1|\mathbf{x}) > P(C_2|\mathbf{x})$$

$$\frac{p(\mathbf{x}|C_1)P(C_1)}{\cancel{p(\mathbf{x})}} > \frac{p(\mathbf{x}|C_2)P(C_2)}{\cancel{p(\mathbf{x})}}$$

$$\ln p(\mathbf{x}|C_1) + \ln P(C_1) > \ln p(\mathbf{x}|C_2) + \ln P(C_2)$$

$$\ln \frac{p(\mathbf{x}|C_1)}{p(\mathbf{x}|C_2)} + \ln \frac{P(C_1)}{P(C_2)} > 0$$

Pravděpodobnostní generativní model

- Modelujme rozložení tříd gaussovskym rozložením:

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

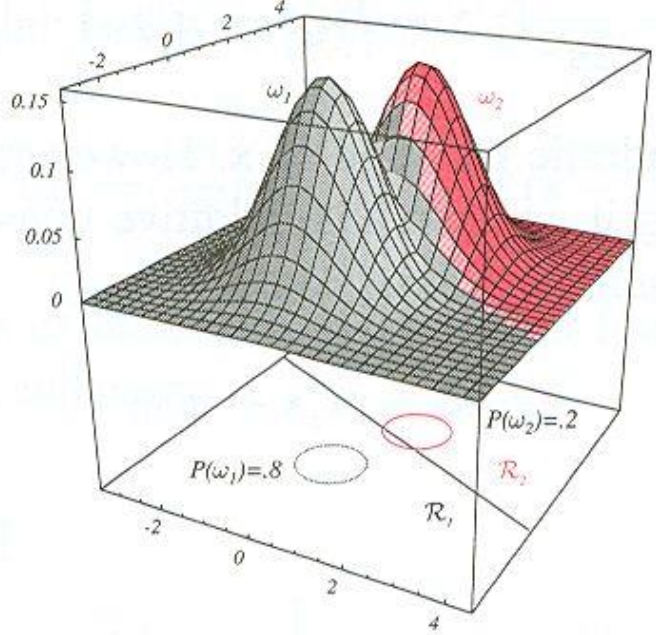
- Pokud náš model omezíme tak, ze každá třída má svou střední $\boldsymbol{\mu}_k$ hodnotu, ale kovarianční matice $\boldsymbol{\Sigma}$ je společná pro obě třídy, tak můžeme psát:

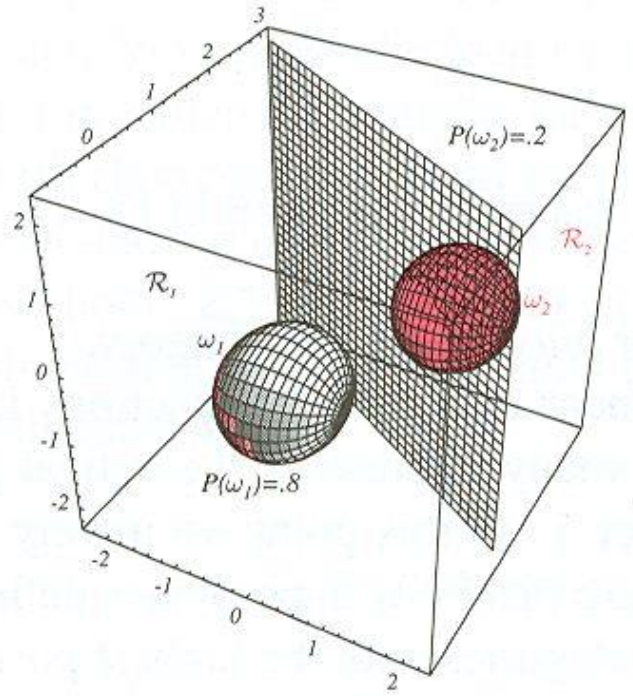
$$y(\mathbf{x}) = \ln\frac{p(\mathbf{x}|C_1)}{p(\mathbf{x}|C_2)} + \ln\frac{P(C_1)}{P(C_2)} = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

kde

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{P(C_1)}{P(C_2)}$$


0.15
0.1
0.05
0
-2
0
2
4
ω_1
ω_2
$P(\omega_2)=.2$
$P(\omega_1)=.8$
$\mathcal{R}_2$
$\mathcal{R}_1$


3
2
1
0
-1
-2
-2
-1
0
1
2
$P(\omega_2)=.2$
$\mathcal{R}_2$
$\mathcal{R}_1$
ω_2
ω_1
$P(\omega_1)=.8$

Maximum likelihood odhad parametrů

- Hledáme parametry modelu

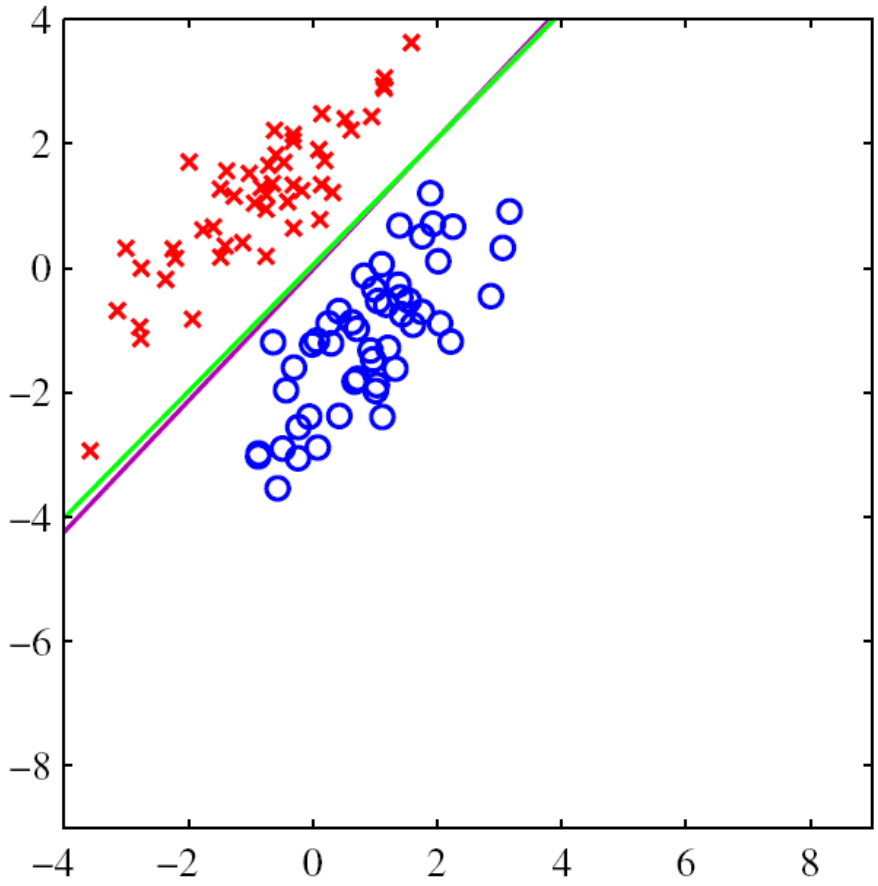
$$\{\widehat{\boldsymbol{\mu}}_1, \widehat{\boldsymbol{\mu}}_2, \widehat{\boldsymbol{\Sigma}}\} = \underset{\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}\}}{\arg\max} \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{t_n}, \boldsymbol{\Sigma})$$

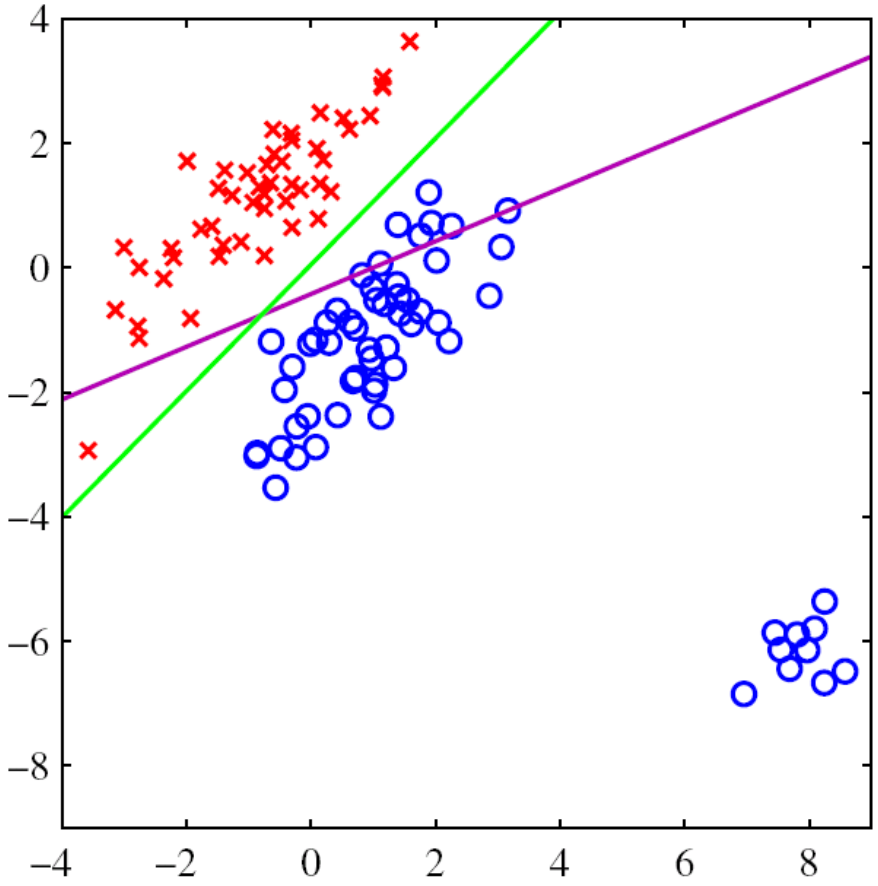
  kde t_n je třída, do které patří vzor $\mathbf{x}_n$ a $\boldsymbol{\mu}_{t_n}$ je střední hodnota této třídy

- Řešením jsou :
  - střední hodnoty $\widehat{\boldsymbol{\mu}}_k$ spočítané z dat jednotlivých tříd
  - kovarianční matice $\boldsymbol{\Sigma}$, která je váhovaným průměrem kovariančních matic $\widehat{\boldsymbol{\Sigma}}_k$ spočtených z dat jednotlivých tříd

$$\widehat{\boldsymbol{\mu}}_k = \frac{1}{N_k} \sum_{n:t_n=k} \mathbf{x}_n \qquad \widehat{\boldsymbol{\Sigma}}_k = \frac{1}{N_k} \sum_{n:t_n=k} (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_k)^{\mathrm{T}}$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{k=1}^{K} N_k \widehat{\boldsymbol{\Sigma}}_k = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_{t_n})(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_{t_n})^{\mathrm{T}}$$

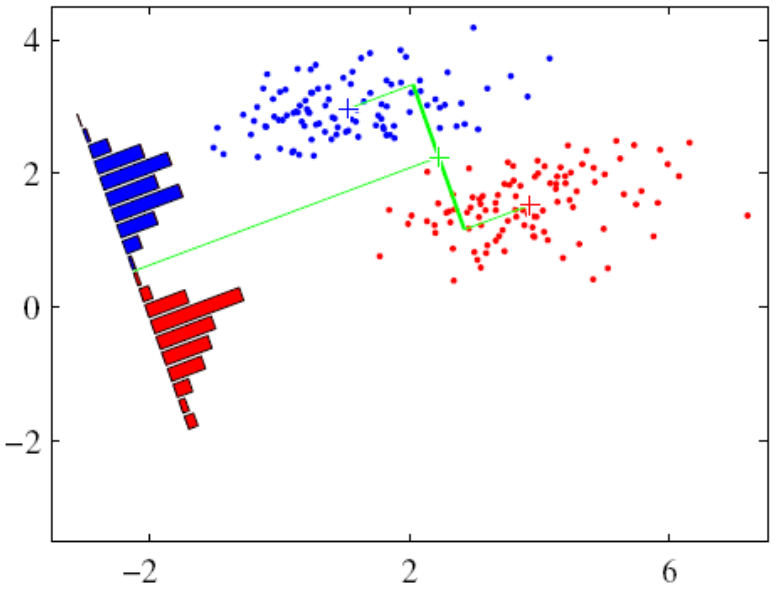




- V případě kdy ovšem naše data nerespektují předpoklad gaussovských rozložení a sdílené kovarianční matice. Klasifikátor může selhat – fialová rozhodovací linie
- Lepší výsledky dostaneme s diskriminativně natrénovaným klasifikátorem, který bude vysvětlen později – zelená rozhodovací linie

Opakování LDA

- Snažíme se data promítnout do takového směru, kde
  - Maximalizujeme vzdálenost mezi středními hodnotami tříd
  - Minimalizujeme průměrnou varianci tříd
- Maximalizujeme tedy



$$m_k = \mathbf{w}^{\mathrm{T}}\mathbf{m}_k$$

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2$$

$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1).$$

- Pro dvě třídy je $\mathbf{w}$ totožné s tím které jsme obdrželi pro náš generativní klasifikátor.
- Generativní klasifikátor ovšem zvolí i práh w_0

Generativní model a zobecněny lineární klasifikátor

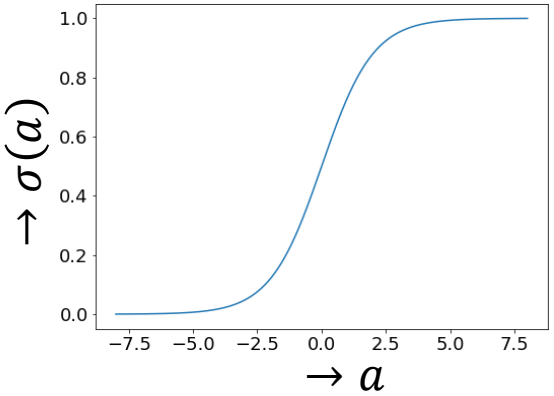
Nyní použijme zobecněný lineární klasifikátor

$$y(\mathbf{x}) = \sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right)$$

kde stále platí, že $\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = \ln\frac{p(\mathbf{x}|C_1)}{p(\mathbf{x}|C_2)} + \ln\frac{P(C_1)}{P(C_2)}$

a kde aktivační funkce je logistická sigmoida

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$



Potom lze hodnotu tohoto zobecněného linearního klasifikátoru přímo interpretovat jako posteriorní pravděpodobnost třídy C_1

$$p(C_1|\mathbf{x}) = \sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right)$$

Jiné generativní lineární klasifikátory

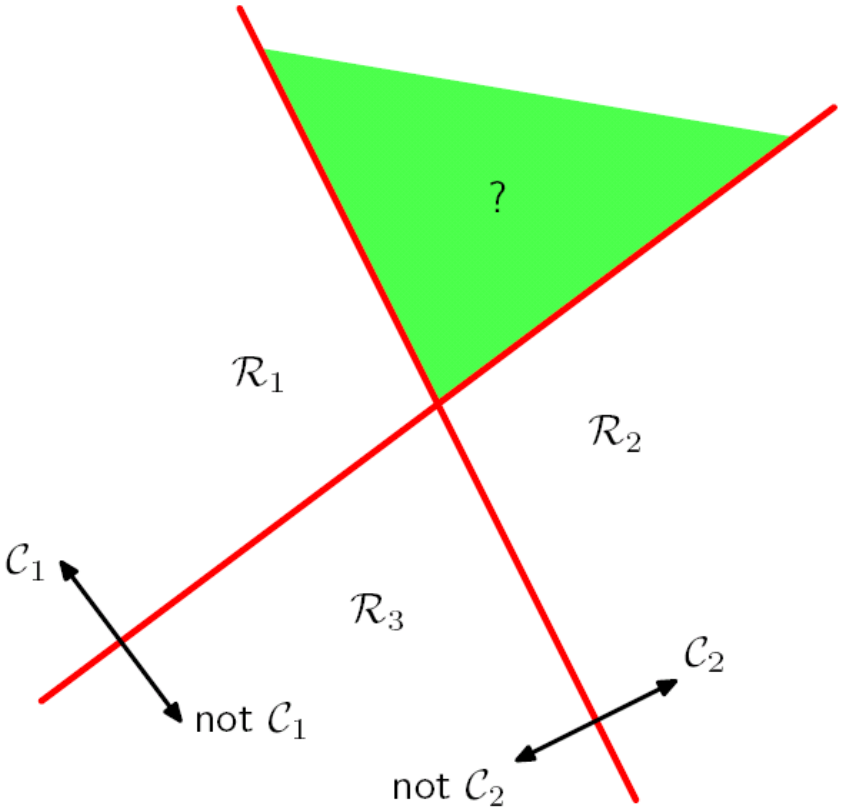
- Lineární klasifikátor dostaneme nejen pro gaussovské rozložení, ale pro celou třídu rozložení s exponencialní rodiny, které lze zapsat v následující formě:

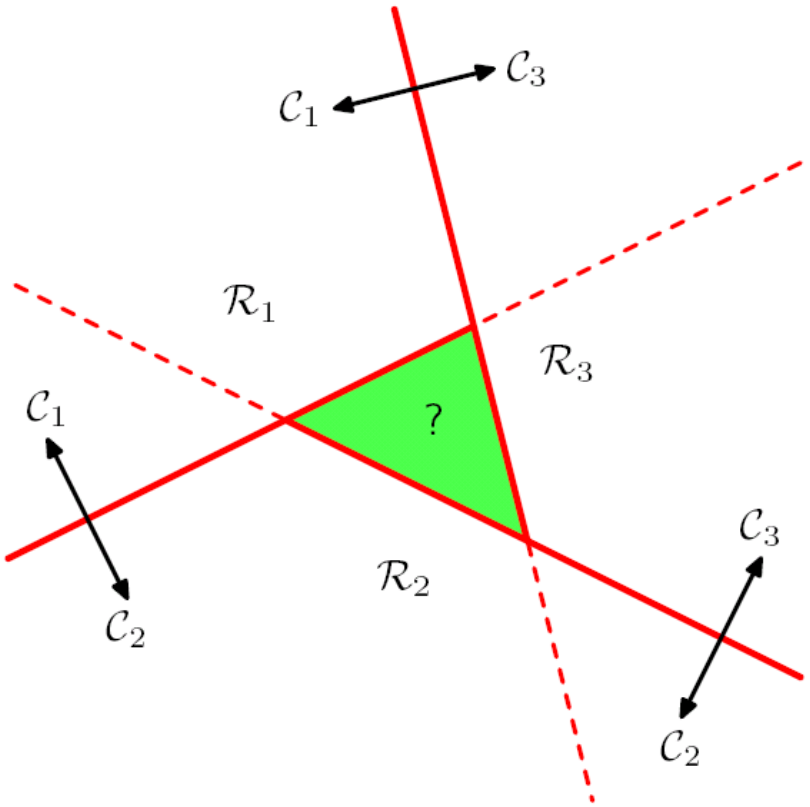
$$p(\mathbf{x}|\boldsymbol{\lambda}_k, s) = \frac{1}{s} h\left(\frac{1}{s}\mathbf{x}\right) g(\boldsymbol{\lambda}_k) \exp\left\{\frac{1}{s}\boldsymbol{\lambda}_k^{\mathrm{T}}\mathbf{x}\right\}$$

kde vektor $\boldsymbol{\lambda}_k$ má každá třída svůj vlastní, zatím co parametr s je sdíleny všemi třídami

Problém s více třídami

- Klasifikace
  - jeden proti všem
  - Každý s každým





Lineární klasifikátor – více tříd

- Nejlépe je mít jednu lineární funkci pro každou třídu k

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

- Vyber třídu s největším $y_k(\mathbf{x})$
- Rozhodovací linie je opět lineární dána

$$y_k(\mathbf{x}) = y_j(\mathbf{x})$$

- Kde *k* a *j* jsou dvě nejpravděpodobnější třídy pro dané **x**

$$(\mathbf{w}_k - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{k0} - w_{j0}) = 0$$

- Pro dvě třídy řešení degraduje k tomu co už jsme viděli

Gaussovský lineární klasifikátor pro více tříd

- Opět modelujme rozložení tříd gaussovskym rozložením:

$$\ln p(\mathbf{x}|C_k) = -\frac{D}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)$$

- Pokud náš model omezíme tak, ze každá třída má svou střední $\boldsymbol{\mu}_k$ hodnotu, ale kovarianční matice $\boldsymbol{\Sigma}$ je společná pro obě třídy, tak můžeme psát:

$$P(C_k|\mathbf{x}) = \frac{P(\mathbf{x}, C_k)}{\sum_l p(\mathbf{x}, C_l)} = \frac{\exp(a_k + \mathrm{const}(\mathbf{x}))}{\sum_l \exp(a_l + \mathrm{const}(\mathbf{x}))} = \frac{\exp(a_k)}{\sum_l \exp(a_l)}$$

funkce softmax

$$\ln p(\mathbf{x}, C_k) = a_k + \mathrm{const}(\mathbf{x})$$
$$a_k = \mathbf{x}^{\mathrm{T}}\mathbf{w}_k + w_{k0}$$
$$\mathbf{w}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$$
$$w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln P(C_k)$$

Konstanta, která nezáleží na třídě. Vykrátila by se ve funkci softmax a tedy ji nemusíme vůbec počítat

Odvození

$$
\begin{aligned}
&\ln p(\mathbf{x}, C_k) \\
&\quad = \ln p(\mathbf{x}|C_k) + \ln p(C_k) \\
&\quad = -\frac{D}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) + \ln p(C_k) \\
&\quad = -\frac{D}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x} + \mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k{}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(C_k) \\
&\quad = \mathrm{const}(\mathbf{x}) + \mathbf{x}^{\mathrm{T}}\mathbf{w}_k + w_{k0} = a_k + \mathrm{const}(\mathbf{x})
\end{aligned}
$$

kde si uvědomíme, že $\mathrm{const}(\mathbf{x})$ závisí na $\mathbf{x}$, ale nezavisí na třídě k,

$$\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k = \boldsymbol{\mu}_k{}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x}$$

a tedy

$$\frac{1}{2}\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \frac{1}{2}\boldsymbol{\mu}_k{}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x} = \mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$$

Softmax funkce

Vstupem softmax funkce je vektor

$$\text{softmax}_k(\mathbf{a}) = \frac{\exp(a_k)}{\sum_l \exp(a_l)} = P(C_k|\mathbf{x})$$

k-tý element výstupu funkce

$$\text{softmax}\left(\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix}\right) = \frac{1}{\sum_l \exp(a_l)} \begin{bmatrix} \exp(a_1) \\ \exp(a_2) \\ \vdots \\ \exp(a_K) \end{bmatrix}$$

Funkce vrací vektor hodnot (pravděpodobností)

Převede vector logaritmu nenormalizovaných pravděpodobností tříd na pravděpodobnosti tříd

$$\text{softmax}\left(\begin{bmatrix} \log P(C_1|\mathbf{x}) + const \\ \log P(C_2|\mathbf{x}) + const \\ \vdots \\ \log P(C_K|\mathbf{x}) + const \end{bmatrix}\right) = \text{softmax}\left(\begin{bmatrix} \log P(\mathbf{x}, C_1) \\ \log P(\mathbf{x}, C_2) \\ \vdots \\ \log P(\mathbf{x}, C_K) \end{bmatrix}\right) = \begin{bmatrix} P(C_1|\mathbf{x}) \\ P(C_2|\mathbf{x}) \\ \vdots \\ P(C_K|\mathbf{x}) \end{bmatrix}$$

Gaussovský lineární klasifikátor pro více tříd II

$$P(C_k|\mathbf{x}) = \frac{P(\mathbf{x}, C_k)}{\sum_l p(\mathbf{x}, C_l)} = \frac{\exp(a_k + \text{const}(\mathbf{x}))}{\sum_l \exp(a_l + \text{const}(\mathbf{x}))} = \frac{\exp(a_k)}{\sum_l \exp(a_l)}$$

$$\ln p(\mathbf{x}, C_k) = a_k + \text{const}(\mathbf{x})$$
$$a_k = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$
$$\mathbf{w}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$$
$$w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln P(C_k)$$

Pravděpodobností všech tříd můžeme tedy efektivně vypočítat jako

$$\begin{bmatrix} P(C_1|\mathbf{x}) \\ P(C_2|\mathbf{x}) \\ \vdots \\ P(C_K|\mathbf{x}) \end{bmatrix} = \text{softmax}\left(\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix}\right) = \text{softmax}(\mathbf{a}) = \text{softmax}(\mathbf{W}\mathbf{x} + \mathbf{w}_0)$$

kde $\mathbf{w}_k^T$ jsou řádky matice $\mathbf{W}$ a w_{k0} jsou koeficienty vektoru $\mathbf{w}_0$.

$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K]^{\mathrm{T}}$ $\quad$ $\mathbf{w}_0 = [w_{10}, w_{20}, \ldots, w_{k0}]^{\mathrm{T}}$

Lineární logistická regrese pro více tříd

$$\begin{bmatrix} P(C_1|\mathbf{x}) \\ P(C_2|\mathbf{x}) \\ \vdots \\ P(C_K|\mathbf{x}) \end{bmatrix} = \mathrm{softmax}\left(\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix}\right) = \mathrm{softmax}(\mathbf{a}) = \mathrm{softmax}(\mathbf{W}\mathbf{x})$$

kde opět předpokládáme $x_0 = 1$ a nemusíme tedy explicitně zavádět w_{k0}.

Nyní budeme parametry $\mathbf{W}$ odhadovat tak, abychom přímo maximalizovali pravděpodobnost anotací $\mathbf{t} = [\, t_1, t_2 \ldots, t_N]^{\mathrm{T}}$, tedy

$$P(\mathbf{t}|\mathbf{X}) = \prod_n P(t_n|\mathbf{x}_n) = \prod_n \mathrm{softmax}_{t_n}(\mathbf{W}\mathbf{x}_{\mathrm{n}}) = \prod_n \frac{\exp(\mathbf{w}_{t_n}^{\mathrm{T}}\mathbf{x}_n)}{\sum_{l=1}^{K}\exp(\mathbf{w}_l^{\mathrm{T}}\mathbf{x}_n)}$$

kde $\mathbf{w}_k^{\mathrm{T}}$ je k-tý řádek matice $\mathbf{W}$ a t_n je index třídy n-tého trénovacího vzoru.

Všiměme si , že maximalizujeme stejnou onjektivní funkci jako pro při odhadu maximalně věrohodných parametrů diskrétního rozložení, jen je teď pravděpodobnost třídy podmíněna pozorováním $\mathbf{x}$.

Lineární logistická regrese – II.

- Místo maximlizování

$$P(\mathbf{t}|\mathbf{X}) = \prod_n P(t_n|\mathbf{x}_n) = \prod_n \mathrm{softmax}_{t_n}(\mathbf{W}\mathbf{x}_\mathrm{n}) = \prod_n \frac{\exp(\mathbf{w}_{t_n}^\mathrm{T}\mathbf{x}_n)}{\sum_{l=1}^K \exp(\mathbf{w}_l^\mathrm{T}\mathbf{x}_n)}$$

lidé ve strojovém učení často mluví o minimalizování ekvivalentí chybové funkce známé jako ***křížová entropie (cross-entropy)***

$$E(\mathbf{w}) = -\ln P(\mathbf{t}|\mathbf{X}) = -\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \ln P(t_n|\mathbf{x}_n) = -\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \ln \frac{\exp(\mathbf{w}_{t_n}^\mathrm{T}\mathbf{x}_n)}{\sum_{l=1}^C \exp(\mathbf{w}_l^\mathrm{T}\mathbf{x}_n)}$$

- Hledáme minimum této funkce, takže derivujeme abychom dostali gradient

$$\nabla_{\mathbf{w}_j} E(\mathbf{W}) = \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \left(\frac{\exp\left(\mathbf{x}_n^\mathrm{T}\mathbf{w}_j\right)}{\sum_l \exp\left(\mathbf{x}_n^\mathrm{T}\mathbf{w}_l\right)} - \delta(t_n = j) \right) \mathbf{x}_n$$

a hledáme takové$\mathbf{w}_j$ (pro všechna j) pro které $\nabla_{\mathbf{w}_j} E(\mathbf{W}) = \mathbf{0}$

Lineární logistická regrese – III.

Gradient muzueme pocitat pro kazdý řadek matice $\mathbf{W}$ zvlašť

$$\nabla_{\mathbf{w}_j} E(\mathbf{W}) = \frac{\partial E(\mathbf{W})}{\partial \mathbf{w}_j} = \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \Big(\mathrm{softmax}_j(\mathbf{W}\mathbf{x}_\mathrm{n}) - \delta(t_n = j)\Big)\,\mathbf{x}_n$$

Nebo můžeme rovnou pořítat derivaci $E(\mathbf{W})$ podle celé matice $\mathbf{W}$

$$\frac{\partial E(\mathbf{W})}{\partial \mathbf{W}} = \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \big(\mathrm{softmax}(\mathbf{W}\mathbf{x}_\mathrm{n}) - \mathrm{onehot}(t_n)\big)\mathbf{x}_n^\mathrm{T} = \ (\mathbf{Y} - \mathbf{T})\,\mathbf{X}^\mathrm{T}$$

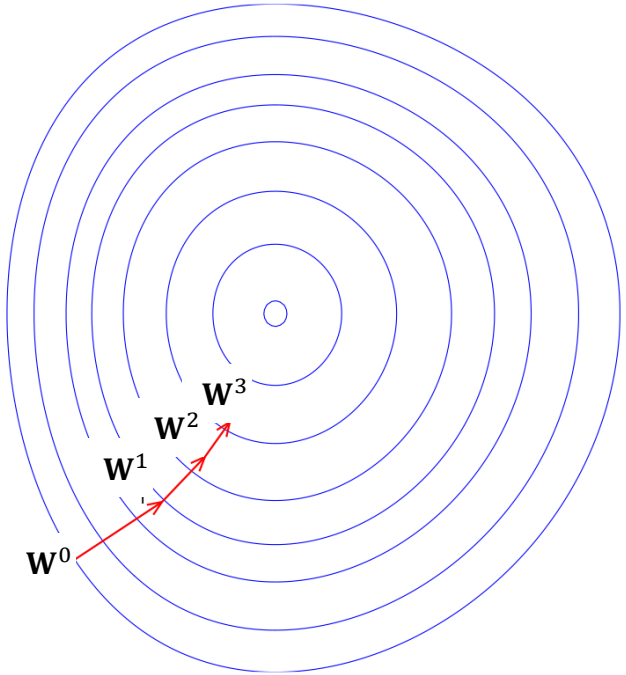
- $\delta(i = j) = 1$ pokud $i = j$ a jinak je 0
- $\mathrm{onehot}(k)$ je vektor nul a pouze k -tý element je 1
- $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ je matice trénovacích vzorů (ve sloupcích)
- $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_N]$ kde sloupce $\mathbf{t}_1 = \mathrm{onehot}(t_\mathrm{n})$ jsou anotace
- $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N]$ kde sloupce jsou predikované pravděpodobnosti tříd $\mathbf{y}_\mathrm{n} = \ \mathrm{softmax}(\mathbf{W}\mathbf{x}_\mathrm{n})$

Metoda gradientního sestupu

- Opakovaně měníme parametry tak, že jimi pohybujeme v malých krocích ve směru opačném ke gradientu $\nabla E(\mathbf{W})$ (tedy z kopce dolů) se nedostaneme do minima funkce kde $\nabla E(\mathbf{W}) = \mathbf{0}$

$$\mathbf{W}^{\tau+1} = \mathbf{W}^{\tau} - \eta \frac{\partial E(\mathbf{W}^{\tau})}{\partial \mathbf{W}}$$

- Učící konstanta (*learning rate)* η určuje jak velké kroky děláme a musí být správně nastavena aby algoritmus konvergoval



- Matematicky správně je gradient vektor, ale naše derivace $\frac{\partial E(\mathbf{W}^{\tau})}{\partial \mathbf{W}}$ je matice, tak abychom ji mohli použít přímo pro upravu matice parametrů $\mathbf{W}$.

softmax pro 2 třídy

$$P(C_1|\mathbf{x}) = \text{softmax}_1\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}\right) = \frac{\exp(a_1)}{\sum_l \exp(a_l)} = \frac{\exp(a_1)}{\exp(a_1) + \exp(a_2)}$$

$$= \frac{1}{1 + \frac{\exp(a_2)}{\exp(a_1)}} = \frac{1}{1 + e^{-(a_1 - a_2)}} = \sigma(a_1 - a_2)$$

- Pro dva vstupy, první výstup softmax_1 „degraduje“ na logistickou sigmoidu rozdílu vstupů
- Pro logistickou regresi se dvěmi třídami:

$$P(C_1|\mathbf{x}) = \text{softmax}_1(\mathbf{W}\mathbf{x}) = \text{softmax}_1\left([\mathbf{w}_1, \mathbf{w}_2]^{\mathrm{T}}\mathbf{x}\right) = \sigma\left((\mathbf{w}_1 - \mathbf{w}_2)^{\mathrm{T}}\mathbf{x}\right) = \sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}\right)$$

- A tedy logisticka pro dvě třídy je jen specialním případem kde $\mathbf{w} = \mathbf{w}_1 - \mathbf{w}_2$

$$P(\mathbf{t}|\mathbf{X}) = \prod_n P(t_n|\mathbf{x}_n) = \prod_n \text{softmax}_{t_n}(\mathbf{W}\mathbf{x}_n)$$

$$P(\mathbf{t}|\mathbf{X}) = \prod_n P(C_1|\mathbf{x}_n)^{t_n} P(C_2|\mathbf{x}_n)^{1-t_n} = \prod_n \sigma\left(\mathbf{x}_n^{\mathrm{T}}\mathbf{w}\right)^{t_n} \left(1 - \sigma(\mathbf{x}_n^T\mathbf{w})\right)^{1-t_n}$$

Lineární logistická regrese – 2 třídy

- Uvažujme opět pravděpodobnostní model, kde
$$P(C_1|\mathbf{x}) = \sigma\left(\mathbf{x}^{\mathrm{T}}\mathbf{w}\right)$$
a pravděpodobnost druhé třídy
$$P(C_2|\mathbf{x}) = 1 - P(C_1|\mathbf{x})$$
- kde opět předpokládáme $x_0 = 1$ a nemusíme tedy explicitně zavádět w_0.
- Nyní budeme parametry $\mathbf{w}$ odhadovat tak, abychom přímo maximalizovali pravděpodobnost anotací $\mathbf{t} = [\, t_1, \, t_2 \ldots, t_N]^{\mathrm{T}}$, tedy
$$P(\mathbf{t}|\mathbf{X}) = \prod_n P(t_n|\mathbf{x}_n)$$
- Pro zjednodušení zápisu předpokládejme, že $t_n = 1$, pokud $\mathbf{x}_n$ paří do třídy C_1 a $t_n = 0$, pokud $\mathbf{x}_n$ paří do třídy C_2.Potom můžeme psát
$$P(\mathbf{t}|\mathbf{X}) = \prod_n P(C_1|\mathbf{x}_n)^{t_n} P(C_2|\mathbf{x}_n)^{1-t_n} = \prod_n \sigma\left(\mathbf{x}_n^{\mathrm{T}}\mathbf{w}\right)^{t_n} \left(1 - \sigma(\mathbf{x}_n^{T}\mathbf{w})\right)^{1-\, t_n}$$
- Všiměme si , že maximalizujeme stejnou onjektivní funkci jako pro pří odhadu maximalně věrohodných parametrů diskrétního rozložení, jen je teď pravděpodobnost třídy podmíněna pozorováním $\mathbf{x}$.

Lineární logistická regrese – 2 třídy – II.

- Místo maximlizování

$$P(\mathbf{t}|\mathbf{X}) = \prod_n P(t_n|\mathbf{x}_n) = \prod_n \sigma\left(\mathbf{x}_n^{\mathrm{T}}\mathbf{w}\right)^{t_n} \left(1 - \sigma(\mathbf{x}_n^{T}\mathbf{w})\right)^{(1-\,t_n)}$$

lidé ve strojovém učení často mluví o minimalizování ekvivalentí chybové funkce známé jako ***křížová entropie (cross-entropy)***

$$E(\mathbf{w}) = -\ln P(\mathbf{t}|\mathbf{X}) = -\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} t_n \ln \sigma\left(\mathbf{x}_n^{\mathrm{T}}\mathbf{w}\right) + (1 - t_n) \ln\left(1 - \sigma\left(\mathbf{x}_n^{\mathrm{T}}\mathbf{w}\right)\right)$$

- Hledáme minimum této funkce, takže derivujeme abychom dostali gradient

$$\nabla E(\mathbf{w}) = \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} (\sigma\left(\mathbf{x}_n^{\mathrm{T}}\mathbf{w}\right) - t_n)\mathbf{x}_n$$

a hledáme takové $\mathbf{w}$ pro které $\nabla E(\mathbf{w}) = \mathbf{0}$

Lineární logistická regrese – 2 třídy – III.

- Pomocí maticového nasobeni mužeme

$$\nabla E(\mathbf{w}) = \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \left(\sigma\left(\mathbf{x}_n^{\mathrm{T}}\mathbf{w}\right) - t_n\right)\mathbf{x}_n$$

přepsat jako

$$\nabla E(\mathbf{w}) = \mathbf{X}(\mathbf{y} - \mathbf{t})$$

kde $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{\mathrm{N}}]$ je matice trenovacích vzorů, $\mathbf{t} = [\, t_1, \, t_2, \ldots, t_N]^{\mathrm{T}}$ sloupcový vektor odpovídajících $(0/1)$ anotací a $\mathbf{y} = [\, y_1, \, y_2 \ldots, y_N]^{\mathrm{T}}$ je sloupcový vektor vystupů klasifikátoru

$$y_n = P(C_1|\mathbf{x}_n) = \sigma\left(\mathbf{x}_n^{\mathrm{T}}\mathbf{w}\right)$$

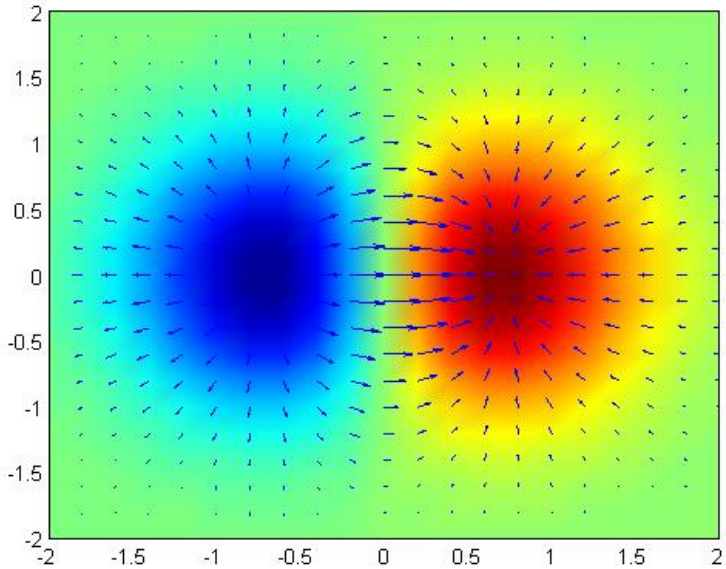
Vyjádření parametrů $\mathbf{w}$ pro $\nabla E(\mathbf{w}) = \mathbf{0}$ bohužel nemá analytické řešení a musíme přistoupit k numerické optimalizaci, např. pomocí metody gradientního sestupu *(gradient descent)*

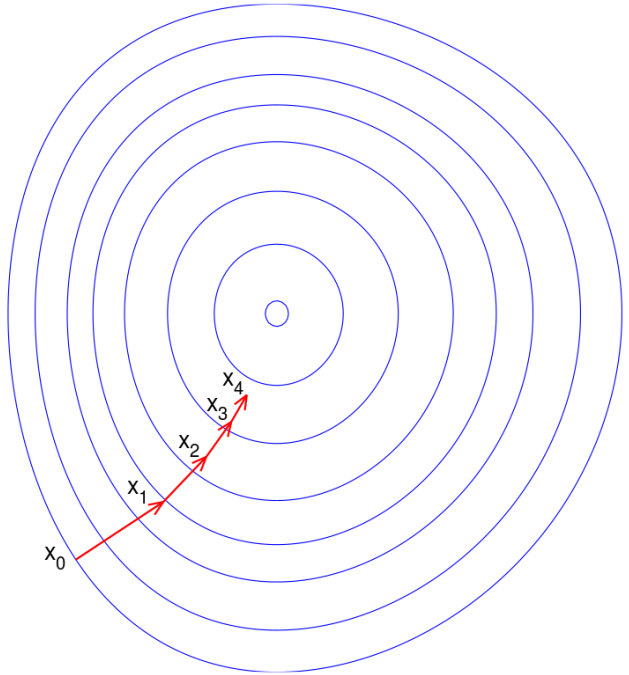
Metoda gradientního sestupu

- Opakovaně měníme parametry tak, že jimi pohybujeme v malých krocích ve směru opačném ke gradientu $\nabla E(\mathbf{w})$ (tedy z kopce dolů) se nedostaneme do minima funkce kde $\nabla E(\mathbf{w}) = \mathbf{0}$

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \eta \nabla E(\mathbf{w}^{\tau})$$

- Učící konstanta (*learning rate)* η určuje jak velké kroky děláme a musí být správně nastavena aby algoritmus konvergoval





Lineární logistická regrese: odhad parametrů

- Rychlejší konvergenci dosáhneme pomocí Newton-Raphson optimalizace:
  - Kolem stávajícího řešení $\mathbf{w}^{\tau}$ aproximujeme chybovou funkci ∇E pomoci Taylorova rozvoje druhého řádu, čímž obdržíme kvadratickou formu (vícerozměrné zobecnění kvadratické funkce).
  - Jako nové řešení zvolíme to, kde má tato kvadratická forma minimum.

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \mathbf{H}(\mathbf{w}^{\boldsymbol{\tau}})^{-1}\nabla E(\mathbf{w}^{\boldsymbol{\tau}})$$

kde $\mathbf{H}(\mathbf{w}^{\boldsymbol{\tau}}) = \mathbf{X}\mathbf{R}\,\mathbf{X}^{\mathrm{T}}$ je matice druhých derivací (Hessian matrix).

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \left(\mathbf{X}\mathbf{R}\,\mathbf{X}^{\mathrm{T}}\right)^{-\mathbf{1}}\mathbf{X}(\mathbf{y} - \mathbf{t})$$

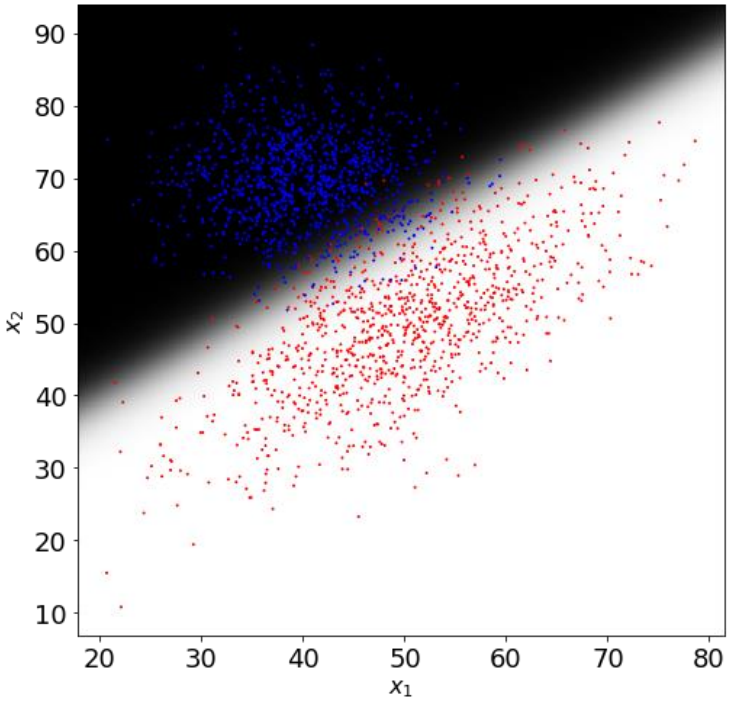
- $\mathbf{R}$ je diagonální matice s diagonálou $\mathrm{diag}(\mathbf{R}) = \mathbf{y}(1 - \mathbf{y})$

- Pozor! Stejně jako i metody gradientního sestupu není zaručeno, že každý krok zlepšení řešení. Metoda může začít divergovat, ale dá se řešit např. (opakovaným) půlením kroku $\mathbf{w}^{\tau+1} := (\mathbf{w}^{\tau+1} + \mathbf{w}^{\tau})/2$.

Logistická regrese – příklad

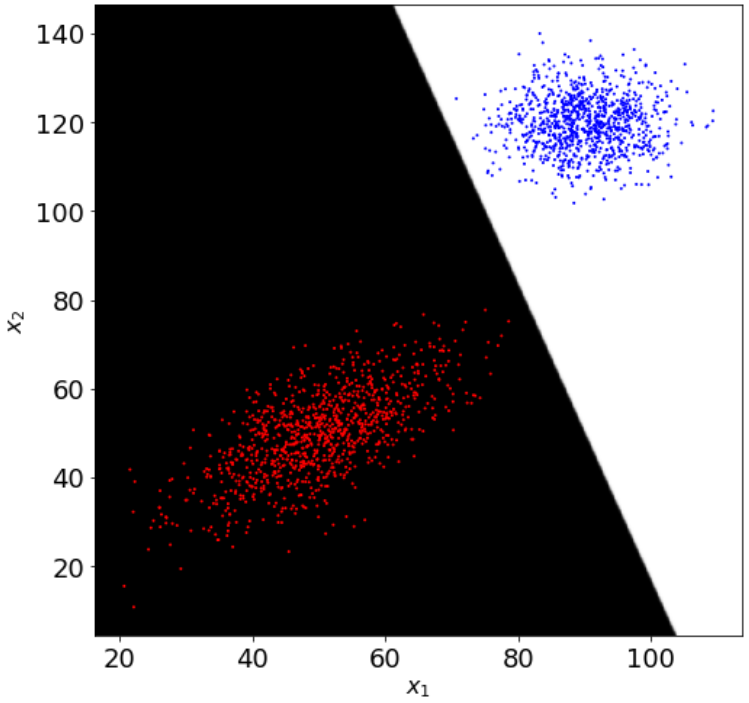
$$P(C_1|\mathbf{x}) = \sigma\left(\mathbf{x}^{\mathrm{T}}\mathbf{w}\right) = \sigma(w_1 x_1 + w_2 x_2 + w_0)$$

Na “přektývajících se třídách” se korektně naučíme odhadovat $P(C_1|\mathbf{x})$



Na “oddělených třídách” může dojít k přetrénování, kde $E(\mathbf{w}) \approx \mathbf{0}$, i když řešení není uspokojivé.

Koeficienty w_1 a w_2 jsou příliš velke ⇒ rychlá změna $P(C_1|\mathbf{x})$



Regularizace parametrů

- Do objektivní funkce přidáme regularizační člen

$$\lambda \|\mathbf{w}\|_2^2 = \lambda \mathbf{w}^{\mathrm{T}} \mathbf{w} = \lambda \sum_{d=1}^{D} w_d^2$$

který pealizuje vysoké hodnoty w_1 a w_2.

$$E(\mathbf{w}) = -\sum_{n=1}^{N} t_n \ln \sigma\left(\mathbf{x}_n^{\mathrm{T}} \mathbf{w}\right) + (1 - t_n) \ln\left(1 - \sigma\left(\mathbf{x}_n^{\mathrm{T}} \mathbf{w}\right)\right) + \lambda \mathbf{w}^{\mathrm{T}} \mathbf{w}$$

- Potom

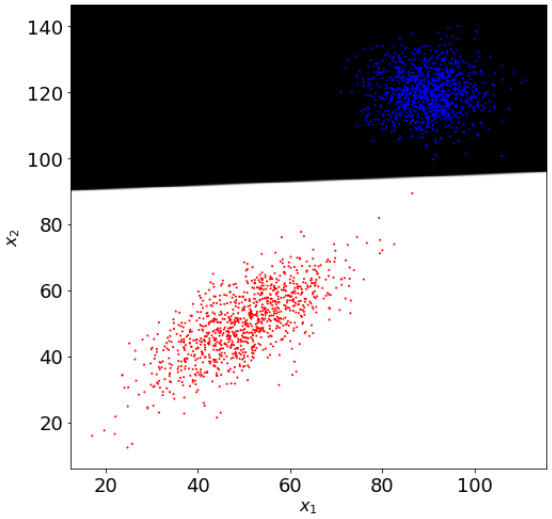
$$\nabla E(\mathbf{w}) = \mathbf{X}(\mathbf{y} - \mathbf{t}) + \lambda\, \mathbf{w} \qquad \mathbf{H}(\mathbf{w}^{\tau}) = \lambda \mathbf{I} + \mathbf{X}\mathbf{R}\,\mathbf{X}^{\mathrm{T}}$$

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \left(\lambda \mathbf{I} + \mathbf{X}\mathbf{R}\,\mathbf{X}^{\mathrm{T}}\right)^{-1} (\mathbf{X}(\mathbf{y} - \mathbf{t}) + \lambda\, \mathbf{w}^{\tau})$$

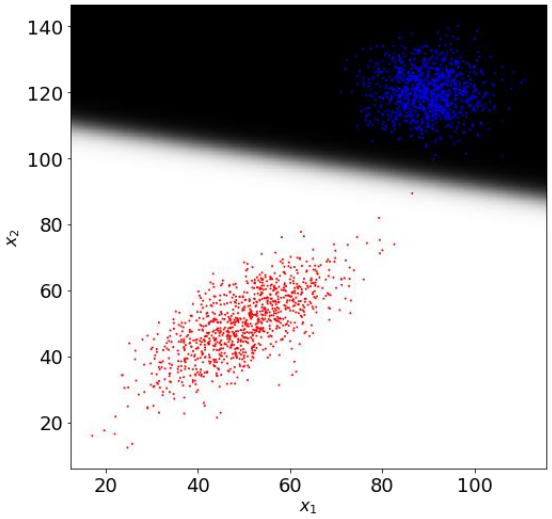
- Zde pro jednoduchost regularizujeme i w_0, ale je lepší to nedělat abycho nestahovali rozhodovací hranici k počátku.

Příklady pro různé hodnoty regularizačního koeficientu λ

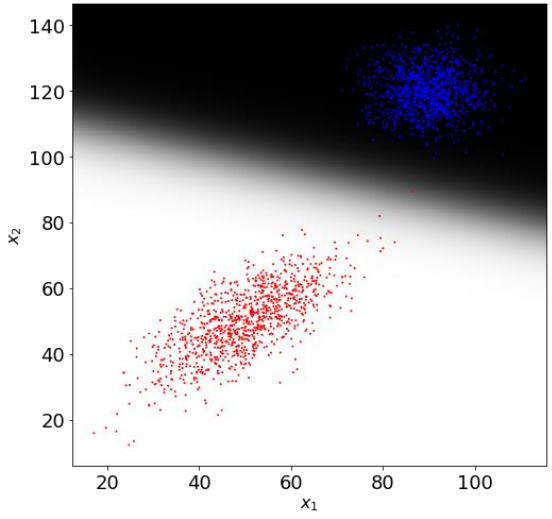
$\lambda = 0$



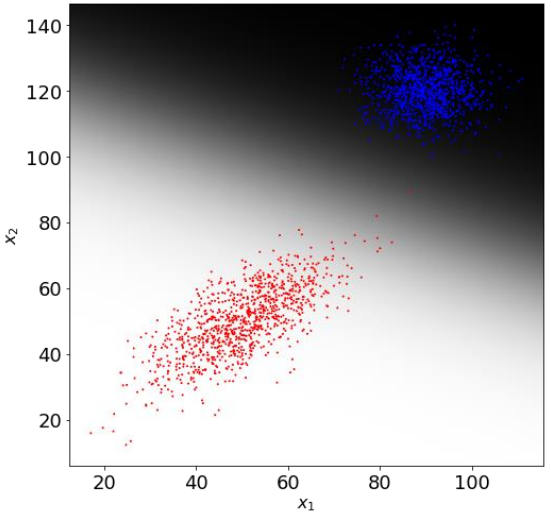
$\lambda = 0.001N$



$\lambda = 0.1N$

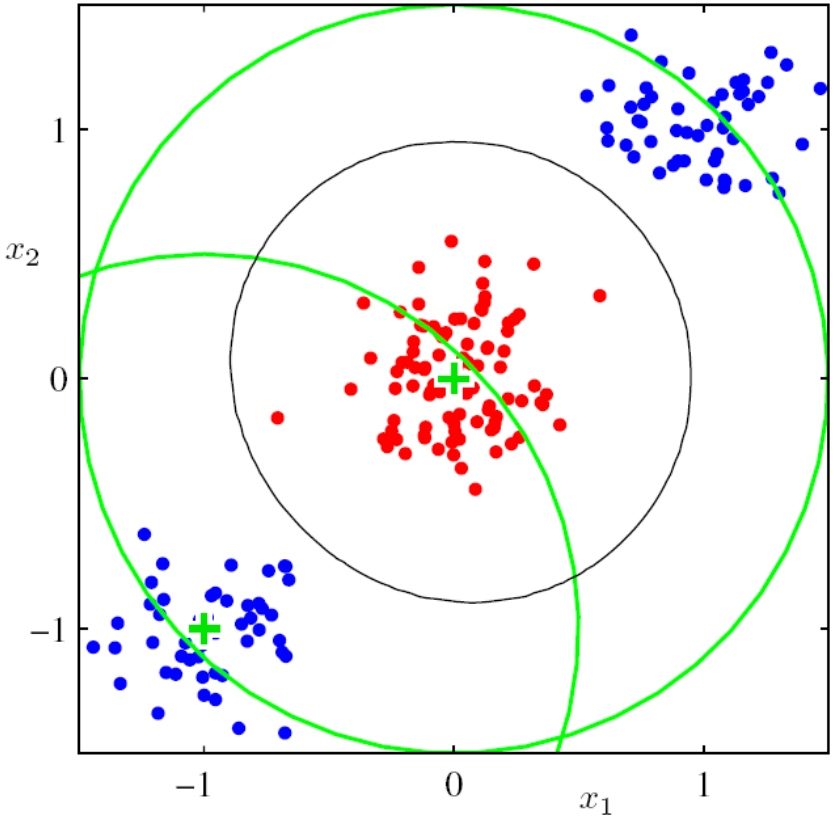


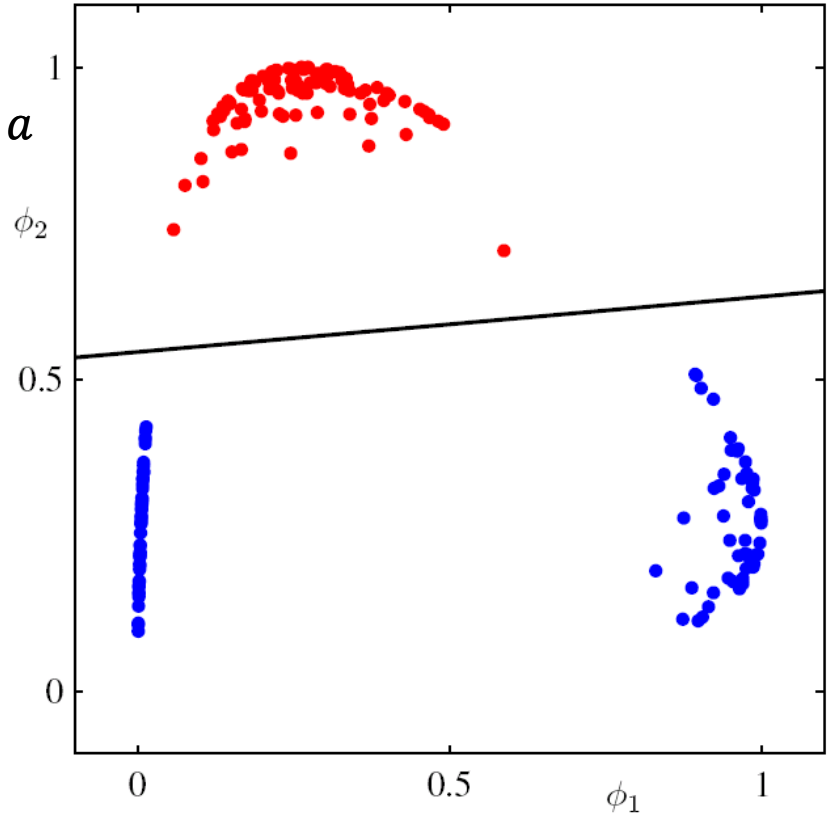
$\lambda = 10N$



Nelineární mapování vstupního vektoru

- Nelze-li původní data lineárně oddělit, možná pomůže jejich nelineární transformace do potenciálně vysocerozměrného prostoru – hlavní myšlenka „kernel methods“ které budou vysvětleny příště
- V našem příkladu pomohlo i mapování dvourozměrných dat do dvou gaussovských funkcí





Lineární logistická regrese: nelineární klasifikace

- Nelineárně transformujeme vstupy $\mathbf{x}$ do vícerozměrného vektoru $\hat{\mathbf{x}}$.
- Jako příklad použijeme trasformaci pomoci polynomů druhého řádu:

$$\hat{\mathbf{x}} = [1, \quad x_1, \quad x_2, \quad x_1 x_2, \quad x_1^2, \quad x_2^2]$$

- Nyní natrénujeme a aplikujeme logistickou regresi nad těmito vícerozměrnými daty. Jakou pravděpodobnost $P(C_1|\mathbf{x})$ odhaduje takový model jako funkci původních dvourozměrných dat $\mathbf{x}$?

- Rozhodovací línie je kuželosečka (jako u gaussovského klasifikátoru).