Vytěžování Dat
Přednáška 9 – Lineární klasifikátor, rozšíření báze, LDA, logistická regrese

Miroslav Čepek

Fakulta Elektrotechnická, ČVUT

Evropský sociální fond Praha & EU:
Investujeme do vaší budoucnosti

25.11.2014

Klasifikátory

- Jaké znáte klasifikační algoritmy?

Klasifikátory

- Jaké znáte klasifikační algoritmy?
  - naivní Bayesův klasifikátor (NB),
  - rozhodovací strom,
  - k nejbližších sousedů (kNN).

Klasifikátory

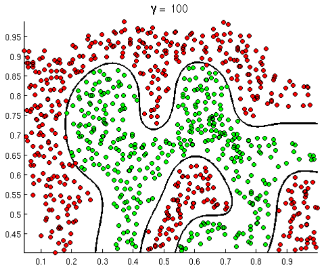
- Jaké znáte klasifikační algoritmy?
  - naivní Bayesův klasifikátor (NB),
  - rozhodovací strom,
  - k nejbližších sousedů (kNN).
- Jak fungují?

Klasifikátory

- Jaké znáte klasifikační algoritmy?
  - naivní Bayesův klasifikátor (NB),
  - rozhodovací strom,
  - k nejbližších sousedů (kNN).
- Jak fungují?
  - NB počítá pravděpodobnost přiřazení do třídy za podmínky dané pozorováním,
  - rozhodovací strom generuje posloupnost rozhodnutí,
  - kNN hledá nejbližší instance z trénovací množiny.

Rozhodovací hranice

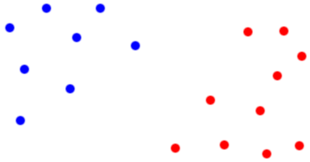
- Další možný pohled je z hlediska **rozhodovací hranice**.
- Rozhodovací hranice je místo, kde instance přestávají patřit do třídy A a začínají patřit do třídy B.



Andrew Ng: Machine Learning. Open Classroom course.

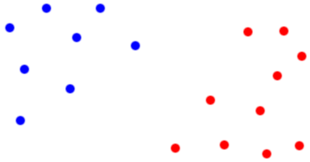
Rozhodovací hranice (2)

- Jak bude vypadat nejjednodušší rozhodovací hranice, která data oklasifikuje bez chyby?



Rozhodovací hranice (2)

- Jak bude vypadat nejjednodušší rozhodovací hranice, která data oklasifikuje bez chyby?



- Stačí přece přímka!
- Nejjednodušší = s nejmenším počtem parametrů.

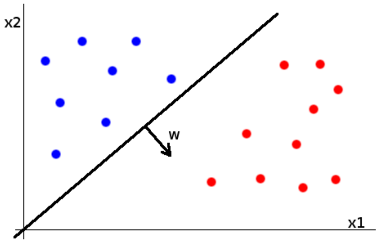
Lineární klasifikátor

- Takovou přímku umíme celkem jednoduše najít a vyrobit :).
- Jak vypadá rovnice přímky, resp. **nadroviny** pro větší dimenze?
- Uvažujme d dimenzionální reálný vstupní prostor $\mathbf{x} \in \mathbb{R}^d$.
- Výstup nechť je binární $y \in \{+1, -1\}$.

Lineární klasifikátor

- Takovou přímku umíme celkem jednoduše najít a vyrobit :).
- Jak vypadá rovnice přímky, resp. **nadroviny** pro větší dimenze?
- Uvažujme d dimenzionální reálný vstupní prostor $\mathbf{x} \in \mathbb{R}^d$.
- Výstup nechť je binární $y \in \{+1, -1\}$.

$$\sum_{j=1}^{d} w_j x_j = \mathbf{w} \cdot \mathbf{x} = 0$$



Lineární klasifikátor (2)

- Jak odhadneme, do které třídy patří vzor $\mathbf{x}$?

Lineární klasifikátor (2)

- Jak odhadneme, do které třídy patří vzor $\mathbf{x}$?
  - Prostým dosazením!
  - Pokud $\mathbf{w} \cdot \mathbf{x} > 0$ vzor patří do třídy $+1$,
  - pokud $\mathbf{w} \cdot \mathbf{x} < 0$ vzor patří do třídy -1.
  - 0 je rozhodovací hranice, kde se nelze rozhodnout.
  - Naštěstí se toto nestává často.
  - Formálně lze zapsat
    $$y' = signum(\mathbf{w} \cdot \mathbf{x})$$
  - kde $signum$ je znaménkovou funkcí
    - vrací $+1$ pro kladná čísla, -1 pro záporná a 0 pro nulový vstup.

Lineární klasifikátor (3)

- Vidíte v rovnici $\mathbf{w} \cdot \mathbf{x} = 0$ nějaký problém?

Lineární klasifikátor (3)

- Vidíte v rovnici $\mathbf{w} \cdot \mathbf{x} = 0$ nějaký problém?
  - Zkusme pro $d = 2$ dosadit do rovnice $x_1 = x_2 = 0$. Co vyjde?
  - 0 bez ohledu na nastavení vah w_1 a w_2.
  - Rozhodovací hranice vždy prochází 0, resp. vektorem $(0, 0, \ldots, 0)$.
  - Nepříjemné omezení – hranicí pouze otáčíme kolem středu. Co s tím?
- Můžeme rozšířit rovnice rozhodovací hranice a klasifikátoru na

$$\sum_{j=1}^{d} w_j x_j + \Theta = 0, \text{ resp. } y' = signum(\mathbf{w} \cdot \mathbf{x} + \Theta)$$

  - Θ je aditivní konstanta a označuje se jako **práh**.

Práh

- Zbývá najít vektor koeficientů (vah) $(w_1, w_2, \ldots, w_n)$ a práh Θ.
- Představuje práh Θ anomálii, která potřebuje speciální zacházení?

Práh

- Zbývá najít vektor koeficientů (vah) $(w_1, w_2, \ldots, w_n)$ a práh Θ.
- Představuje práh Θ anomálii, která potřebuje speciální zacházení?
- Tak napůl :). Nepotřebuje, pokud trochu upravíme vstupní data.
- Co kdyby všechny vzory měly jeden atribut (vstupní proměnnou, řekněme x_0) se stejnou hodnotou?
- Když pak budu počítat $w_0 x_0$, vyjde mi vždy stejná hodnota . . . to je přece náš práh!

Práh

- Zbývá najít vektor koeficientů (vah) $(w_1, w_2, \ldots, w_n)$ a práh Θ.
- Představuje práh Θ anomálii, která potřebuje speciální zacházení?
- Tak napůl :). Nepotřebuje, pokud trochu upravíme vstupní data.
- Co kdyby všechny vzory měly jeden atribut (vstupní proměnnou, řekněme x_0) se stejnou hodnotou?
- Když pak budu počítat $w_0 x_0$, vyjde mi vždy stejná hodnota ... to je přece náš práh!
- Ke každému vzoru přidám ještě jednu dimenzi s konstantní hodnotou – pro jednoduchost 1.
- Pak už se o práh nemusím speciálně starat.

Gradientní sestup – krok stranou

- Gradientní sestup je iterativní optimalizační metoda.
  - Hledáme minimum funkce $f(\mathbf{x})$.
  - Tj. hledáme takové $\mathbf{x}$, kde má funkce $f(\mathbf{x})$ nejnižší hodnotu.
  - Mám nějaké $\mathbf{x_0}$ a znám hodnotu $f(\mathbf{x_0})$. Hledám v okolí $\mathbf{x_0}$, nejaké $\mathbf{x_1}$, kde bude hodnota $f(\mathbf{x_1})$ nižší. A to budu opakovat stále dokola, až to dál nepůjde.

Gradientní sestup – krok stranou

- Gradientní sestup je iterativní optimalizační metoda.
  - Hledáme minimum funkce $f(\mathbf{x})$.
  - Tj. hledáme takové $\mathbf{x}$, kde má funkce $f(\mathbf{x})$ nejnižší hodnotu.
  - Mám nějaké $\mathbf{x_0}$ a znám hodnotu $f(\mathbf{x_0})$. Hledám v okolí $\mathbf{x_0}$, nejaké $\mathbf{x_1}$, kde bude hodnota $f(\mathbf{x_1})$ nižší. A to budu opakovat stále dokola, až to dál nepůjde.
- "Na horách se chci dostat do údolí. Podívám se, kterým směrem se kopec nejvíc snižuje a udělám krok tím směrem. A zase se podívám, kam je to nejvíc z kopce a udělám krok. A tak dále, až jednou zjistím, že to dál nejde, čili jsem v údolí."

Gradientní sestup – krok stranou (2)

- V matematice existuje možnost, jak zjistit směr největšího vzestupu hodnoty funkce – gradient. Tou je první parciální derivace podle všech dimenzí. Když půjdu přesně opačně, půjdu ve směru největšího poklesu.
- Gradient je vektor parciálních derivací podle jednotlivých proměnných:

$$grad(f(x_1, x_2, \ldots, x_n)) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right)$$

Chybová funkce a gradientní sestup

- Jako u jiných klasifikačních metod hledáme nastavení klasifikátoru (zde vah) takové, abychom správně označili co nejvíc vzorů.
- Uvažujme nejprve chybovou funkci $J(\mathbf{w}, \chi)$, která vrací počet špatně klasifikovaných vzorů $\mathbf{x}$ z množiny vzorů χ (například z trénovací množiny) při daných vahách $\mathbf{w}$.

Chybová funkce a gradientní sestup

- Jako u jiných klasifikačních metod hledáme nastavení klasifikátoru (zde vah) takové, abychom správně označili co nejvíc vzorů.
- Uvažujme nejprve chybovou funkci $J(\mathbf{w}, \chi)$, která vrací počet špatně klasifikovaných vzorů $\mathbf{x}$ z množiny vzorů χ (například z trénovací množiny) při daných vahách $\mathbf{w}$.
  - Nehodí se – neumím spočítat parciální derivace (a ani gradient).

Chybová funkce a gradientní sestup

- Jako u jiných klasifikačních metod hledáme nastavení klasifikátoru (zde vah) takové, abychom správně označili co nejvíc vzorů.
- Uvažujme nejprve chybovou funkci $J(\mathbf{w}, \chi)$, která vrací počet špatně klasifikovaných vzorů $\mathbf{x}$ z množiny vzorů χ (například z trénovací množiny) při daných vahách $\mathbf{w}$.
  - Nehodí se – neumím spočítat parciální derivace (a ani gradient).
- Zkusme jinou

$$E(\mathbf{w}) = \sum_{\mathbf{x_e} \in \chi^{err}} (-y_e \mathbf{w}(\mathbf{t}) \cdot \mathbf{x_e})$$

- kde χ^{err} jsou špatně klasifikované vzory z χ,
- a y_e je korektní klasifikace instance $\mathbf{x_e}$.

Chybová funkce a gradientní sestup

- Jako u jiných klasifikačních metod hledáme nastavení klasifikátoru (zde vah) takové, abychom správně označili co nejvíc vzorů.
- Uvažujme nejprve chybovou funkci $J(\mathbf{w}, \chi)$, která vrací počet špatně klasifikovaných vzorů $\mathbf{x}$ z množiny vzorů χ (například z trénovací množiny) při daných vahách $\mathbf{w}$.
  - Nehodí se – neumím spočítat parciální derivace (a ani gradient).
- Zkusme jinou

$$E(\mathbf{w}) = \sum_{\mathbf{x_e} \in \chi^{err}} (-y_e \mathbf{w}(\mathbf{t}) \cdot \mathbf{x_e})$$

- kde χ^{err} jsou špatně klasifikované vzory z χ,
- a y_e je korektní klasifikace instance $\mathbf{x_e}$.
  - To už je spojitá funkce.
  - E mohu minimalizovat změnou vah – pomocí gradientního sestupu.

Chybová funkce a gradientní sestup

- Parciální derivace chybové funkce E podle jedné proměnné:

$$\frac{\partial E}{\partial w_1} = \frac{\partial \sum_{\mathbf{x_e} \in \chi^{err}} (-y_e w_1(t) x_{1e})}{\partial w_1}$$

- Kde x_{1e} je první souřadnice z $\mathbf{x_e}$ a w_1 je první souřadnice z $\mathbf{w}$.

$$\frac{\partial \sum_{x_1 \in \chi^{err}} (-y_e w_1(t) x_{1e})}{\partial w_1} = -\sum_{\mathbf{x_e} \in \chi^{err}} y_e x_{1e}$$

- Chybu nejvíce snížím změnou váhy w_1 o součet hodnot prvních vstupních proměnných x_1 pozitivních vzorů, na kterých jsem udělal chybu (u negativních vzorů budu odečítat).
- Analogicky i ostatní vstupní proměnné (včetně prahu).

Učení lineárního klasifikátoru

- **Perceptronový algoritmus**
  - Založen přesně na této myšlence ...

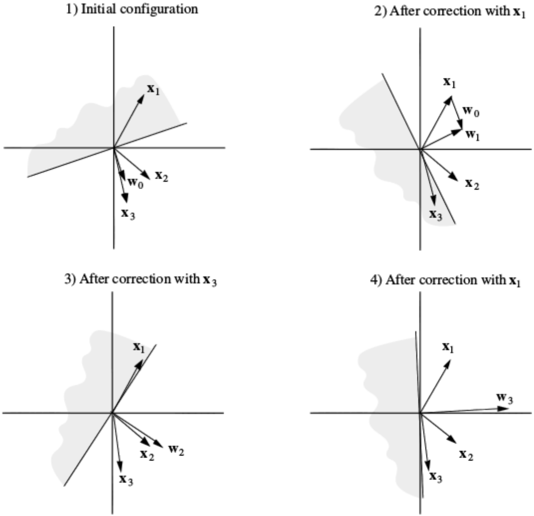
1. Inicializuj váhy na náhodné hodnoty.
2. Vezmi náhodný vstupní vzor $\mathbf{x}$. A spočítej hodnotu

$$y'(\mathbf{x}) = sgn(\sum_j w_j x_j = \mathbf{w} \cdot \mathbf{x})$$

3. Liší-li se požadovaný výstup y od odhadu y', oprav váhy $\mathbf{w}$ takto:
   - $e(\mathbf{x_e}) = signum(y_e - y'_e)$
   - $\mathbf{w}(t+1) = \mathbf{w}(t) + \eta e(\mathbf{x_e})\mathbf{x_e}$
   - η je koeficient učení, který s postupem času může klesat k nule.
4. Pokud chceš, pokračuj bodem 2).

Ilustrace učení percetronu

- 4 pozitivní příklady, dělící přímka středem . . . ukázka konvergence.



Rojas: Neural Networks.

Jednoznačnost výsledné přímky

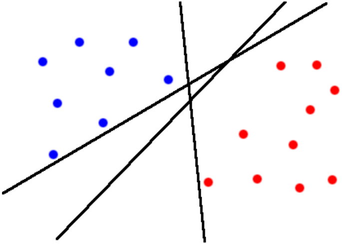
- Jak poznám, že se perceptronový algoritmus zastavil?

Jednoznačnost výsledné přímky

- Jak poznám, že se perceptronový algoritmus zastavil?
- Je výsledná přímka jediná možná?

Jednoznačnost výsledné přímky

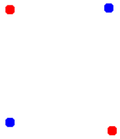
- Jak poznám, že se perceptronový algoritmus zastavil?
- Je výsledná přímka jediná možná?
- Ne, takových přímek je dokonce nekonečně mnoho.
- Přesto jsou některé lepší něž jiné.



- Kterou mám vybrat? A kterou nám vybere Perceptronový algoritmus?

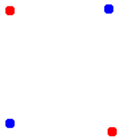
Lineárně separovatelné problémy

- Lze pomocí perceptronu bezchybně klasifikovat následující data?



Lineárně separovatelné problémy

- Lze pomocí perceptronu bezchybně klasifikovat následující data?



- Ne!
- Třídám (datům), které lze oddělit přímkou, říkáme lineárně separovatelná.

Lineárně ne–separovatelné problémy

- Lze pomocí lineárního klasifikátoru vyřešit lineárně ne-separovatelné problémy?

Lineárně ne–separovatelné problémy

- Lze pomocí lineárního klasifikátoru vyřešit lineárně ne-separovatelné problémy?
- Ne. K tomu potřebuji přidat další kvalitu.

Lineárně ne–separovatelné problémy

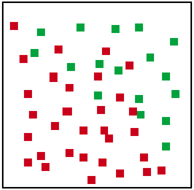
- Lze pomocí lineárního klasifikátoru vyřešit lineárně ne-separovatelné problémy?
- Ne. K tomu potřebuji přidat další kvalitu.
  1. Jednou z možností je zkombinovat lineární klasifikátory.
     O tom bude příští přednáška.

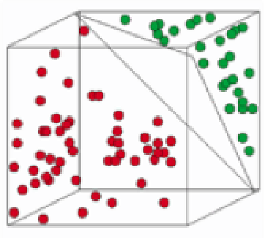
Lineárně ne–separovatelné problémy

- Lze pomocí lineárního klasifikátoru vyřešit lineárně ne-separovatelné problémy?
- Ne. K tomu potřebuji přidat další kvalitu.
  1. Jednou z možností je zkombinovat lineární klasifikátory.
     O tom bude příští přednáška.
  2. Rozšíření báze.

Rozšíření báze

- Když data nejsou oddělitelná v původním prostoru, zkusme jestli by nebyla lineárně separovatelná ve více-dimenzionálním prostoru.





Transformace

- Zase existuje mnoho transformací.
- Přidej ke stávajícím dimenzím **součiny originálních dimenzí**.
  - Jedna z těch jednodušších metod.
  - Pracuje s parametrem s – maximálním stupněm součinu.

Transformace

- Zase existuje mnoho transformací.
- Přidej ke stávajícím dimenzím **součiny originálních dimenzí**.
  - Jedna z těch jednodušších metod.
  - Pracuje s parametrem s – maximálním stupněm součinu.
- Příklad pro $s = 2$ a původní dimenze (x_1, x_2)
  - získám 5D prostor dimenzí $(x_1, x_2, \chi_1, \chi_2, \chi_3)$
  - $\chi_1 = x_1^2$
  - $\chi_2 = x_1 * x_2$
  - $\chi_3 = x_2^2$

Učení nelineárního klasifikátoru

- Jak naučím klasifikátor v novém prostoru?

Učení nelineárního klasifikátoru

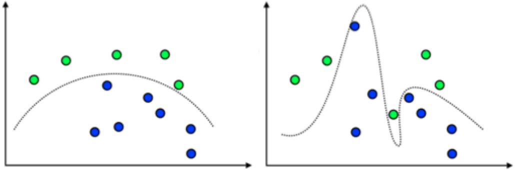
- Jak naučím klasifikátor v novém prostoru?
- Díky rozšíření báze nakládám s více-dimenzionálním prostorem a (třeba) dokáži třídy lineárně separovat.
- Postupuji úplně stejně jako v předchozím případě. Tj. použiji např. perceptronový algoritmus.

Učení nelineárního klasifikátoru

- Jak naučím klasifikátor v novém prostoru?
- Díky rozšíření báze nakládám s více-dimenzionálním prostorem a (třeba) dokáži třídy lineárně separovat.
- Postupuji úplně stejně jako v předchozím případě. Tj. použiji např. perceptronový algoritmus.
- Pokud se mi nepodaří získat lineární separaci, mohu zkusit zvýšit maximální stupeň součinu a zkusit separaci znovu.
- Obecně platí, že k oddělení N bodů stačí prostor o $N-1$ dimenzích.

Rozhodovací hranice

- Jak pak bude vypadat projekce rozhodovací hranice zpět do originálních dimenzí?



- Podařilo se dosáhnout "ohnutí" rozhodovací hranice.
- Hranice bude "poddajnější" s rostoucím s.

Rozhodovací hranice (2)

- Můžu zvyšovat maximální stupeň součinu (parametr s) do nekonečna? Nebo někde narazím?

Rozhodovací hranice (2)

- Můžu zvyšovat maximální stupeň součinu (parametr s) do nekonečna? Nebo někde narazím?
- S vyšším maximálním stupněm s vyvstávají dva problémy:
  1. Musím nastavovat více vah,
     perceptronový (nebo jakýkoliv jiný) algoritmus to nemusí zvládnout,
     váhy se nastaví špatně.
  2. Zvyšuji pravděpodobnost, že se klasifikátor přeučí,
     tj. naučí se šum a chyby v trénovacích datech,
     nedokáže vlastnosti trénovacích dat zobecnit,
     podobně jako například 1-NN klasifikátor.

Klasifikace do několika tříd

- Co když mám více než dvě třídy?
- Mohu použít (ne-)lineární separaci? Případně jak?

Klasifikace do několika tříd

- Co když mám více než dvě třídy?
- Mohu použít (ne-)lineární separaci? Případně jak?
- Budu rozhodovat o každé třídě samostatně.
  - Pro problém s N třídami budu mít N klasifikátorů.
  - Každý bude říkat – "Instance patří resp. **ne**patří do mé třídy".

Klasifikace do několika tříd

- Co když mám více než dvě třídy?
- Mohu použít (ne-)lineární separaci? Případně jak?
- Budu rozhodovat o každé třídě samostatně.
  - Pro problém s N třídami budu mít N klasifikátorů.
  - Každý bude říkat – "Instance patří resp. **ne**patří do mé třídy".
- Při rozhodování o neznámé instanci $\mathbf{x}$ se zeptám všech N klasifikátorů na jejich rozhodnutí:
  1. $\mathbf{x}$ patří právě do jedné třídy $\rightarrow$ OK.
  2. $\mathbf{x}$ patří do více tříd $\rightarrow$ rozhodu např. podle hodnoty $\mathbf{w} \cdot \mathbf{x}$ (čím je větší, tím je $\mathbf{x}$ dále od rozhodovací hranice).
  3. analogicky, pokud $\mathbf{x}$ nepatří do žádné třídy $\rightarrow$ hledám třídu, kam $\mathbf{x}$ nepatří nejméně :).

Problémy lineárních klasifikátorů

- Linearita
- Pokud se dostatečně pohnu i jen v jedné proměnné, můžu získat obrácený výsledek.
- Hodnota $\mathbf{w} \cdot \mathbf{x}$ roste se vzdáleností od rozhodovací hranice a nemá žádnou interpretaci.
- Lineární klasifikátory špatně snášejí chybějící hodnoty.
- Pokud provádím rozšíření báze, může trvat učení velmi dlouho.

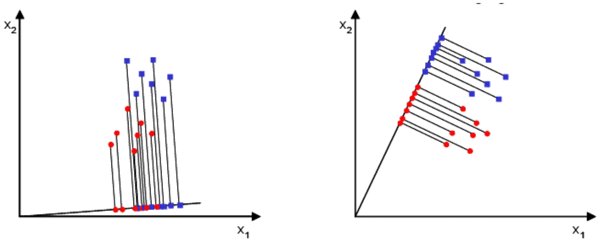
Linear Discriminant Analysis

- LDA je statistický přístup k problému lineární separace.
- Pamatujete si PCA?
  - Principal Component Analysis – mluvili jsme o ní u SOM.

Linear Discriminant Analysis

- LDA je statistický přístup k problému lineární separace.
- Pamatujete si PCA?
  - Principal Component Analysis – mluvili jsme o ní u SOM.
- LDA redukuje původní prostor na přímku (1 dimenzi)
  - Dělá to tak, aby co nejvíce oddělila vzory různých tříd.
  - V jistém smyslu je to opak rozšíření báze.

Linear Discriminant Analysis (2)



- Obraz vzoru v nové souřadnici získám jako $\mathbf{w} \cdot \mathbf{x}$, kde $\mathbf{w}$ jsou váhy.
- V principu počítám to samé jako u lineárního klasifikátoru
  - interpretace a postup jsou úplně jiné.

Linear Discriminant Analysis (3)

- Při počítání LDA se snažím maximalizovat "separaci" mezi třídami.
- Nejprve pro každou třídu musím spočítat rozptyl v obraze (scatter).
  - Projekce objektu do nové souřadnice:

$$y_i = \mathbf{x_i} \cdot \mathbf{w}$$

  - Odhad střední hodnoty (pro pozitivní třídu χ^+):

$$\tilde{\mu}_{\chi^+} = \frac{1}{\|\chi^+\|} \sum_{\forall x_i \in \chi^+} y_i = \frac{1}{\|\chi^+\|} \sum_{\forall x_i \in \chi^+} \mathbf{x_i} \cdot \mathbf{w}$$

  - Odhad rozptylu v obraze (pro pozitivní třídu χ^+)

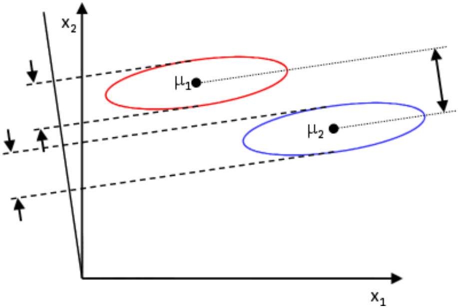
$$\tilde{s}^2_{\chi^+} = \sum_{\forall x_1 \in \chi^+} y_i - \tilde{\mu}_{\chi^+}$$

Linear Discriminant Analysis (4)

- Nyní můžu spočítat míru "separace" pro projekci danou vahami $\mathbf{w}$:

$$J(\mathbf{w}) = \frac{\|\tilde{\mu}_{\chi^+} - \tilde{\mu}_{\chi^-}\|^2}{\tilde{s}^2_{\chi^+} + \tilde{s}^2_{\chi^-}}$$

- Hledáme tedy projekci, která co nejvíc seskupí vzory stejné třídy a zároveň co nejvíc oddělí vzory různých tříd.

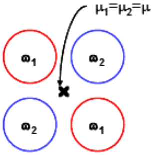


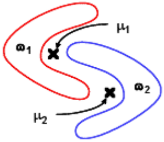
LDA pro více tříd

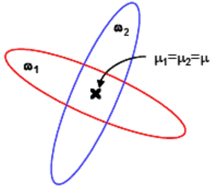
- LDA lze zobecnit tak, aby dokázala klasifikovat do více tříd.
- Pak se nevytváří pouze jedna nová dimenze, ale "počet tříd"-1 nových dimenzí a mírně se upraví vzorečky :).

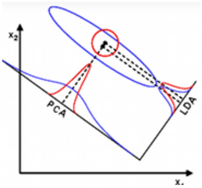
Problémy LDA

- Stejně jako všechno, má i LDA nějaké mouchy:
  - LDA dimenze nemusí stačit ke správném zařazení do třídy.
  - LDA předpokládá normální rozložení dat ve třídách. A selhává, pokud tomu tak není (např. pokud třídy mají komplexní tvary).
  - Rozíly mezi třídami jsou spíše v rozptylu hodnot než v rozdílu středních hodnot.







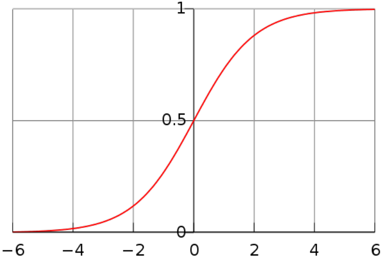


Logistická regrese

- Mohu interpretovat vzdálenosti objektů od rozhodovací hranice?
  - Ideálně jako pravděpodobnost příslušnosti k dané třídě.

Logistická regrese

- Mohu interpretovat vzdálenosti objektů od rozhodovací hranice?
  - Ideálně jako pravděpodobnost příslušnosti k dané třídě.
- Existuje **logistická funkce**
  - Definovaná jako $f(z) = \frac{1}{1+e^{-z}}$.
  - Její hodnoty jsou omezené na interval $0 \dots 1$.
  - Navíc kolem $z = 0$ (rozhodovací hranice) je její změna největší.



Logistická regrese (2)

- Zkusíme zavést pojem $sance = \frac{p}{1-p}$

Logistická regrese (2)

- Zkusíme zavést pojem $sance = \frac{p}{1-p}$
- Zlogaritmujeme ji a položíme rovnou vzdálenosti k dělící hranici
  - Tím nejprve získáme funkci $logit(p) = log(sance(p)) = log(\frac{p}{1-p})$
  - Následně $logit(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k$
  - Kde $\beta_1, \ldots, \beta_k$ jsou regresní koeficienty a β_0 je posun.
  - Přičtení vzdálenosti teď odpovídá násobení šance.

Logistická regrese (2)

- Zkusíme zavést pojem $sance = \frac{p}{1-p}$
- Zlogaritmujeme ji a položíme rovnou vzdálenosti k dělící hranici
  - Tím nejprve získáme funkci $logit(p) = log(sance(p)) = log(\frac{p}{1-p})$
  - Následně $logit(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k$
  - Kde $\beta_1, \ldots, \beta_k$ jsou regresní koeficienty a β_0 je posun.
  - Přičtení vzdálenosti teď odpovídá násobení šance.
- Zkusme zpětně dopočítat pravděpodobnost vzoru $\mathbf{x}$:

$$p(\mathbf{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k)}}$$

- Pro dvě třídy stačí rozhodnutí dle testu $p(\mathbf{x}) > 0.5$.

Shrnutí

- Tři lineární metody klasifikace
  - Perceptron – diskriminativní model, optimalizuje dělící hranici.
  - LDA – vhodná při splnění silných předpokladů, generativní model.
  - Logistická regrese – pravděpodobnostní přístup.
- Na začátku jsme nezmínili použití metody nejmenších čtverců
  - Klasický přístup v lineární regresi.
  - Při klasifikaci pro každou třídu zavedeme jako závislou veličinou indikátorový vektor třídy.
  - Objekt zařazen do třídy, k jejíž nadrovině má největší kladnou vzdálenost.
  - Je citlivá na odlehlé hodnoty.
  - Při větším počtu tříd některé z nich "odmaskovává".
  - Přednost má logistická regrese.

Zdroje

- Perceptron, čtěte např. Hagan et al.: Neural Network Design, `hagan.okstate.edu/4_Perceptron.pdf`
- LDA převzata z `research.cs.tamu.edu/prism/lectures/pr/pr_l10.pdf`
- Logistic Regression tutorial `http://www.omidrouhani.com/research/logisticregression/html/logisticregression.htm`