

Optimální rozdělující nadplocha. Kvadratické programování. Kernel trik. Support vector machines.

Petr Pošík

Katedra kybernetiky
ČVUT FEL

Obsah

Optimální rozdělující
nadplocha

Optimální rozdělující nadplocha

Kvadratické
programování

Kvadratické programování

Kernel trik

Kernel trik

Závěr

Závěr

Optimální rozdělující
nadplocha

Motivace

Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Optimální rozdělující nadplocha

Optimální rozdělovací
nadplocha

Motivace

Optimální rozdělovací
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

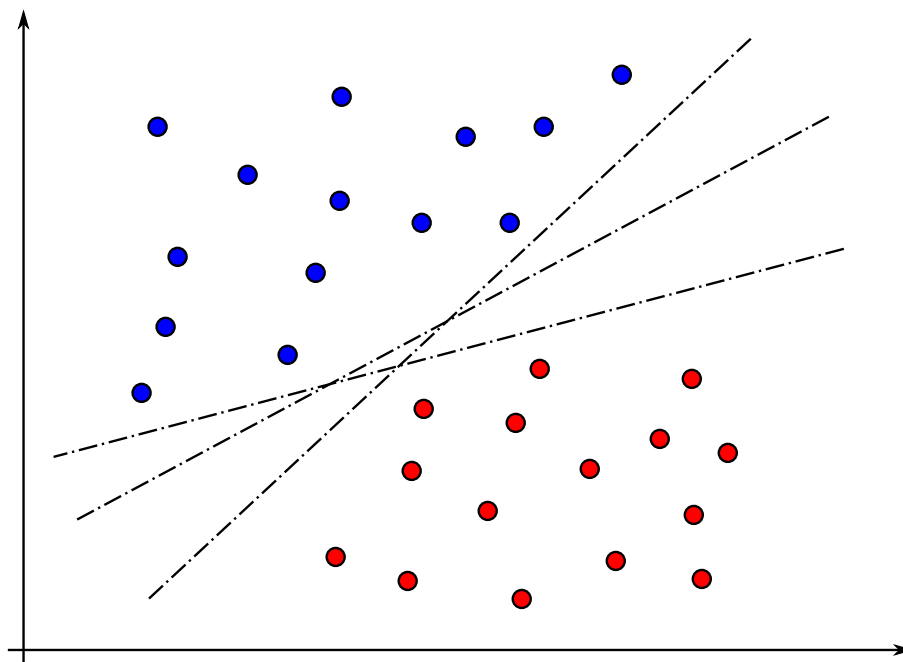
Kvadratické
programování

Kernel trik

Závěr

Perceptronový algoritmus pro lineárně separabilní data

- ✓ najde rozdělovací nadplochu, pokud existuje
- ✓ pokud existuje jedna nadplocha, existuje jich nekonečně mnoho



- ✓ najde *jakoukoliv* rozdělovací nadplochu!

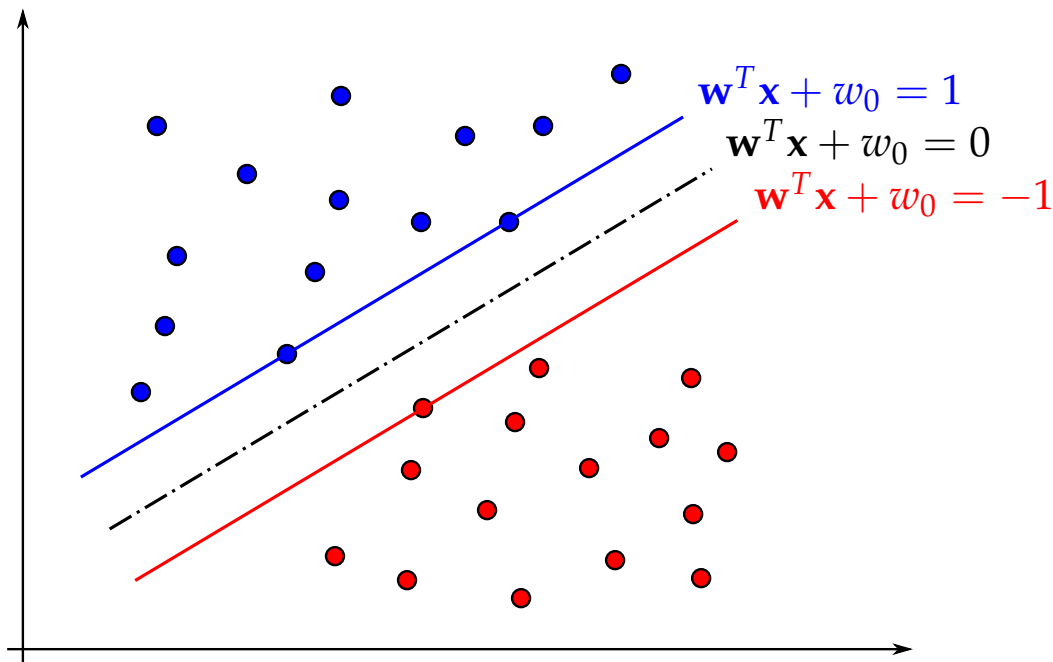
Která z nekonečného množství rozdělovacích nadploh je optimální?

Optimální rozdělující nadplocha

Margin (odstup):

- ✓ šířka pásma, v němž se hranice mezi třídami může pohybovat (kolmo ke své směrnici), aniž by se dotkla některého bodu

Lineární klasifikátor s maximálním marginem



Plus-rovina: $\{x : \mathbf{w}^T \mathbf{x} + w_0 = 1\}$

Minus-rovina: $\{x : \mathbf{w}^T \mathbf{x} + w_0 = -1\}$

Hranice mezi třídami: $\{x : \mathbf{w}^T \mathbf{x} + w_0 = 0\}$

Support vectors:

- ✓ datové body ležící v plus-rovině nebo minus-rovině
- ✓ pouze ty mají vliv na hranici mezi třídami

Proč maximalizovat margin?

- ✓ Intuitivně se to zdá být nejbezpečnější
- ✓ Pokud bychom udělali malou chybu při stanovování hranice, máme nejmenší šanci, že způsobíme špatnou klasifikaci
- ✓ Model je stabilní vůči změně trénovací množiny, pokud nedojde ke změně některého ze support vectors
- ✓ Existují teoretické výsledky (založené na VC dimenzi), že hledat klasifikátor s maximálním marginem je dobré
- ✓ Maximální margin v praxi dobře funguje

Optimální rozdělující
nadplocha

Motivace
Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Jak vypočítat velikost marginu M ze známých w a w_0 ?

Optimální rozdělující
nadplocha

Motivace
Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Jak vypočítat velikost marginu M ze známých $\mathbf{w}$ a w_0 ?

✓ vektor $\mathbf{w}$ je kolmý na plus-rovinu a minus-rovinu.

Optimální rozdělující
nadplocha

Motivace
Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Jak vypočítat velikost marginu M ze známých $\mathbf{w}$ a w_0 ?

- ✓ vektor $\mathbf{w}$ je kolmý na plus-rovinu a minus-rovinu.
Proč?

Optimální rozdělující
nadplocha

Motivace
Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Jak vypočítat velikost marginu M ze známých $\mathbf{w}$ a w_0 ?

- ✓ vektor $\mathbf{w}$ je kolmý na plus-rovinu a minus-rovinu.

Proč? Necht' jsou $\mathbf{u}$ a $\mathbf{v}$ body ležící v plus-rovině. Čemu se rovná $\mathbf{w}^T(\mathbf{u} - \mathbf{v})$?

Optimální rozdělující
nadplocha

Motivace
Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Jak vypočítat velikost marginu M ze známých $\mathbf{w}$ a w_0 ?

- ✓ vektor $\mathbf{w}$ je kolmý na plus-rovinu a minus-rovinu.
- ✓ Necht' $\mathbf{x}^-$ je bod ležící v minus-rovině.
- ✓ Necht' $\mathbf{x}^+$ je bod ležící v plus-rovině, který nejbližší k bodu $\mathbf{x}^-$.
- ✓ Pak $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$.

Optimální rozdělující
nadplocha

Motivace
Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Jak vypočítat velikost marginu M ze známých $\mathbf{w}$ a w_0 ?

- ✓ vektor $\mathbf{w}$ je kolmý na plus-rovinu a minus-rovinu.
- ✓ Necht' $\mathbf{x}^-$ je bod ležící v minus-rovině.
- ✓ Necht' $\mathbf{x}^+$ je bod ležící v plus-rovině, který nejbližší k bodu $\mathbf{x}^-$.
- ✓ Pak $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$.
Proč?

Optimální rozdělující
nadplocha

Motivace
Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Jak vypočítat velikost marginu M ze známých $\mathbf{w}$ a w_0 ?

- ✓ vektor $\mathbf{w}$ je kolmý na plus-rovinu a minus-rovinu.
- ✓ Necht' $\mathbf{x}^-$ je bod ležící v minus-rovině.
- ✓ Necht' $\mathbf{x}^+$ je bod ležící v plus-rovině, který nejbližší k bodu $\mathbf{x}^-$.
- ✓ Pak $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$.
Proč? Úsečka vedoucí z $\mathbf{x}^-$ do $\mathbf{x}^+$ je kolmá na obě roviny a $\mathbf{w}$ je jejím směrovým vektorem.
Abychom se dostali z bodu $\mathbf{x}^-$ do $\mathbf{x}^+$, musíme urazit λ -násobek vektoru $\mathbf{w}$.

Velikost marginu

Optimální rozdělující
nadplocha

Motivace
Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Jak vypočítat velikost marginu M ze známých $\mathbf{w}$ a w_0 ?

- ✓ vektor $\mathbf{w}$ je kolmý na plus-rovinu a minus-rovinu.
- ✓ Necht' $\mathbf{x}^-$ je bod ležící v minus-rovině.
- ✓ Necht' $\mathbf{x}^+$ je bod ležící v plus-rovině, který nejblíže k bodu $\mathbf{x}^-$.
- ✓ Pak $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$.

Víme, že:

Můžeme odvodit:

$$\begin{array}{ll} \mathbf{w}^T \mathbf{x}^+ + w_0 = 1 & \longrightarrow \mathbf{w}^T (\mathbf{x}^- + \lambda \mathbf{w}) + w_0 = 1 \\ \mathbf{w}^T \mathbf{x}^- + w_0 = -1 & \longrightarrow \\ \mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w} & \longrightarrow \mathbf{w}^T \mathbf{x}^- + w_0 + \lambda \mathbf{w}^T \mathbf{w} = 1 \\ M = |\mathbf{x}^+ - \mathbf{x}^-| & \longrightarrow -1 + \lambda \mathbf{w}^T \mathbf{w} = 1 \\ & \longrightarrow \lambda = \frac{2}{\mathbf{w}^T \mathbf{w}} = \frac{2}{|\mathbf{w}|^2} \end{array}$$

Velikost marginu:

$$M = |\mathbf{x}^+ - \mathbf{x}^-| = |\lambda \mathbf{w}| = \lambda |\mathbf{w}| = \frac{2}{|\mathbf{w}|^2} |\mathbf{w}| = \frac{2}{|\mathbf{w}|} \quad (1)$$

Učení klasifikátoru s maximálním marginem

Optimální rozdělující
nadplocha

Motivace

Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Známe-li w a w_0 , můžeme

- ✓ určit, zda jsou všechny trénovací body ve správném poloprostoru, a
- ✓ vypočítat velikost marginu.

Optimální rozdělující
nadplocha

Motivace
Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Známe-li $\mathbf{w}$ a w_0 , můžeme

- ✓ určit, zda jsou všechny trénovací body ve správném poloprostoru, a
- ✓ vypočítat velikost marginu.

Takže nyní nám už jenom zbývá napsat program, který prohledá prostor všech možných $\mathbf{w}$ a w_0 a nalezne takové hodnoty, pro něž výsledný klasifikátor bude korektně klasifikovat všechny trénovací body a bude mít maximální margin.

Optimální rozdělující
nadplocha

Motivace
Optimální rozdělující
nadplocha

Velikost marginu

Učení klasifikátoru s
maximálním marginem

Kvadratické
programování

Kernel trik

Závěr

Známe-li $\mathbf{w}$ a w_0 , můžeme

- ✓ určit, zda jsou všechny trénovací body ve správném poloprostoru, a
- ✓ vypočítat velikost marginu.

Takže nyní nám už jenom zbývá napsat program, který prohledá prostor všech možných $\mathbf{w}$ a w_0 a nalezne takové hodnoty, pro něž výsledný klasifikátor bude korektně klasifikovat všechny trénovací body a bude mít maximální margin.

Jak to udělat?

- ✓ Gradientní metody?
- ✓ Newtonova metoda?
- ✓ Metoda Nelder-Meada?
- ✓ Simulované žíhání?
- ✓ Genetické algoritmy?

Optimální rozdělující
nadplocha

**Kvadratické
programování**

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v
Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Kvadratické programování

Učení pomocí kvadratického programování

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data
QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v
Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

QP je dobře prostudovaná třída optimalizačních algoritmů, které umí hledat extrémy kvadratické funkce reálných proměnných při lineárních omezovacích podmínkách.

Najdi

$$\mathbf{u}^{\text{OPT}} = \arg \min_{\mathbf{u}} \frac{1}{2} \mathbf{u}^T \mathbf{A} \mathbf{u} + \mathbf{b}^T \mathbf{u} + c$$

s P omezujícími podmínkami ve tvaru nerovností a Q omezujícími podmínkami ve tvaru rovností:

$$p_{11}u_1 + \dots + p_{1D}u_D + p_{10} \geq 0$$

$$\vdots$$

$$p_{P1}u_1 + \dots + p_{PD}u_D + p_{P0} \geq 0$$

$$q_{11}u_1 + \dots + q_{1D}u_D + q_{10} = 0$$

$$\vdots$$

$$q_{Q1}u_1 + \dots + q_{QD}u_D + q_{Q0} = 0$$

Existují algoritmy, které řeší tento problém efektivně.

Pravděpodobně byste ale nechtěli takovou metodu sami implementovat.

QP pro maximální margin

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data
QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$.

- ✓ Jaké bude naše kvadratické optimalizační kritérium?
- ✓ Kolik omezení budeme mít?
- ✓ Jak budou vypadat?

QP pro maximální margin

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data
QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$.

- ✓ Jaké bude naše kvadratické optimalizační kritérium?

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0} \mathbf{w}^T \mathbf{w}$$

- ✓ Kolik omezení budeme mít?

- ✓ Jak budou vypadat?

QP pro maximální margin

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data
QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$.

- ✓ Jaké bude naše kvadratické optimalizační kritérium?

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0} \mathbf{w}^T \mathbf{w}$$

- ✓ Kolik omezení budeme mít?

Stejně, jako je trénovacích bodů, tj. N .

- ✓ Jak budou vypadat?

QP pro maximální margin

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data
QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$.

- ✓ Jaké bude naše kvadratické optimalizační kritérium?

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0} \mathbf{w}^T \mathbf{w}$$

- ✓ Kolik omezení budeme mít?

Stejně, jako je trénovacích bodů, tj. N .

- ✓ Jak budou vypadat?

$$\begin{aligned} \mathbf{w}^T \mathbf{x}_i + w_0 &\geq 1 & \text{iff} & & y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + w_0 &\leq -1 & \text{iff} & & y_i = -1 \end{aligned}$$

Lineárně neseparovatelná data

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v
Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Body z trénovací množiny nejsou lineárně separovatelné (třídy se částečně překrývají).
Co dělat?

Lineárně neseparovatelná data

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v
Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Body z trénovací množiny nejsou lineárně separovatelné (třídy se částečně překrývají).
Co dělat?

- ✓ Nápad 1: minimalizovat $\mathbf{w}^T \mathbf{w}$ a současně minimalizovat počet špatně zaklasifikovaných trénovacích příkladů.

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Body z trénovací množiny nejsou lineárně separovatelné (třídy se částečně překrývají).
Co dělat?

- ✓ Nápad 1: minimalizovat $\mathbf{w}^T \mathbf{w}$ a současně minimalizovat počet špatně zaklasifikovaných trénovacích příkladů.

Hmmm, to by se jednalo o multikriteriální optimalizaci. Rozšiřujeme-li margin, zvyšujeme počet špatně zaklasifikovaných trénovacích bodů, a naopak. Špatně definovaný optimalizační problém.

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Body z trénovací množiny nejsou lineárně separovatelné (třídy se částečně překrývají).
Co dělat?

- ✓ Nápad 1: minimalizovat $\mathbf{w}^T \mathbf{w}$ a současně minimalizovat počet špatně zaklasifikovaných trénovacích příkladů.

Hmmm, to by se jednalo o multikriteriální optimalizaci. Rozšiřujeme-li margin, zvyšujeme počet špatně zaklasifikovaných trénovacích bodů, a naopak. Špatně definovaný optimalizační problém.

- ✓ Nápad 1.1: Spojíme obě kritéria do jednoho a jejich vzájemnou významnost určíme parametrem. Budeme minimalizovat $\mathbf{w}^T \mathbf{w} + C(\text{počet špatně zaklasifikovaných trén. příkladů})$.

Optimální rozdělující nadplocha

Kvadratické programování

Učení pomocí kvadratického programování

QP pro maximální margin

Lineárně neseparovatelná data

QP pro neseparabilní data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Body z trénovací množiny nejsou lineárně separovatelné (třídy se částečně překrývají).
Co dělat?

- ✓ Nápad 1: minimalizovat $\mathbf{w}^T \mathbf{w}$ a současně minimalizovat počet špatně zaklasifikovaných trénovacích příkladů.

Hmmm, to by se jednalo o multikriteriální optimalizaci. Rozšiřujeme-li margin, zvyšujeme počet špatně zaklasifikovaných trénovacích bodů, a naopak. Špatně definovaný optimalizační problém.

- ✓ Nápad 1.1: Spojíme obě kritéria do jednoho a jejich vzájemnou významnost určíme parametrem. Budeme minimalizovat $\mathbf{w}^T \mathbf{w} + C(\text{počet špatně zaklasifikovaných trén. příkladů})$.

Nelze vyjádřit jako QP. Nedělá rozdíl mezi kritickými chybami a téměř správnou klasifikací.

Lineárně neseparovatelná data

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

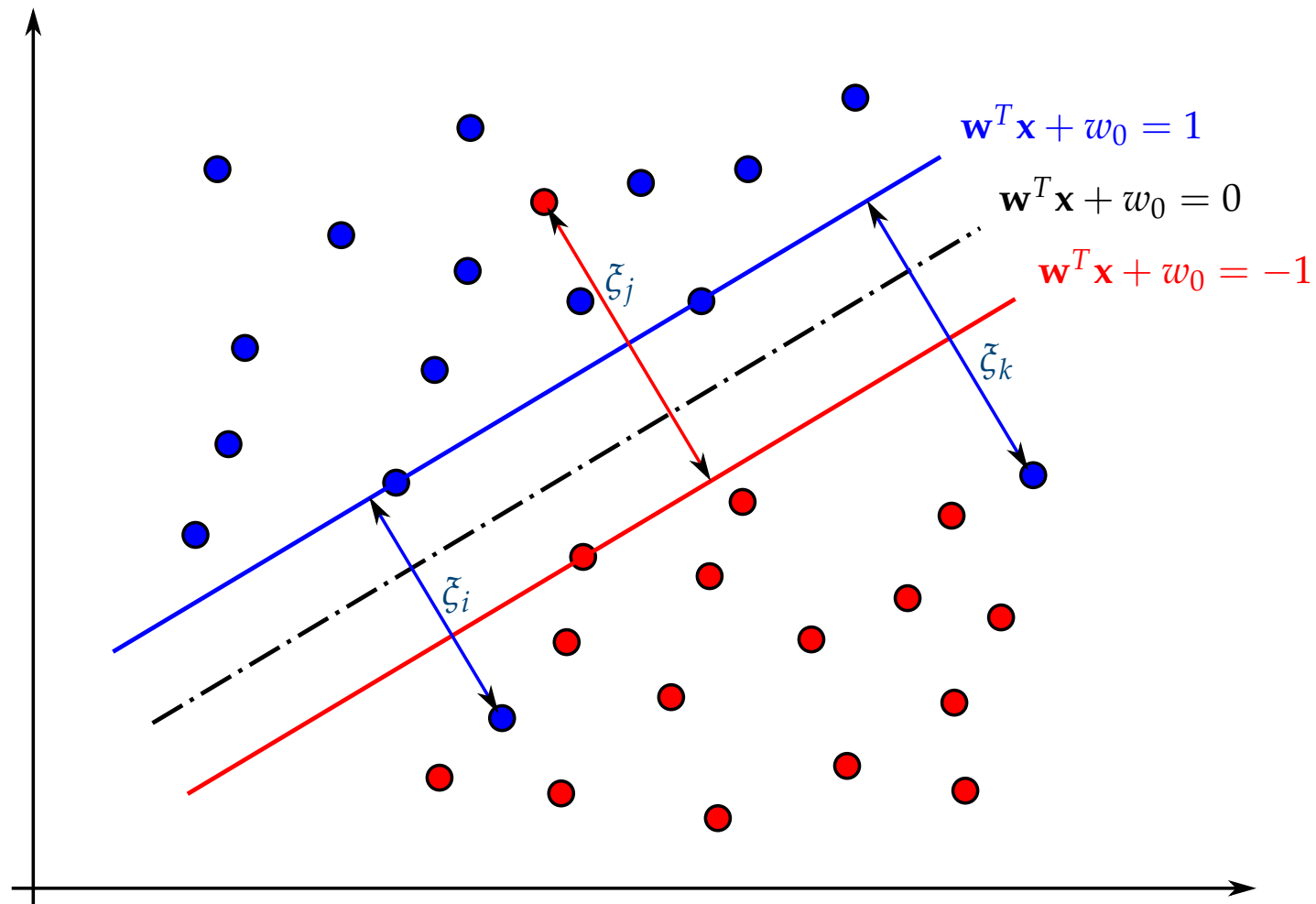
Proč duální formulace?

Kernel trik

Závěr

Body z trénovací množiny nejsou lineárně separovatelné (třídy se částečně překrývají).
Co dělat?

- ✓ Nápad 2.0: minimalizovat $\mathbf{w}^T \mathbf{w} + C \cdot$ (součet vzdáleností špatně zaklasifikovaných trén. příkladů k jejich správnému umístění).



QP pro neseparabilní data

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v
Lagrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$ a minimalizovat součet vzdáleností špatně zaklasifikovaných trén. příkladů k jejich správnému umístění.

✓ Jak budou vypadat omezení?

✓ Jaké bude naše kvadratické optimalizační kritérium?

✓ Kolik omezení budeme mít?

QP pro neseparabilní data

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$ a minimalizovat součet vzdáleností špatně zaklasifikovaných trén. příkladů k jejich správnému umístění.

- ✓ Jak budou vypadat omezení? Zavedeme tzv. *slack variables* ξ_i :

$$\left. \begin{array}{ll} \mathbf{w}^T \mathbf{x}_i + w_0 \geq 1 - \xi_i & \text{iff } y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + w_0 \leq -1 + \xi_i & \text{iff } y_i = -1 \end{array} \right\} y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i \geq 0$$

- ✓ Jaké bude naše kvadratické optimalizační kritérium?

- ✓ Kolik omezení budeme mít?

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$ a minimalizovat součet vzdáleností špatně zaklasifikovaných trén. příkladů k jejich správnému umístění.

- ✓ Jak budou vypadat omezení? Zavedeme tzv. *slack variables* ξ_i :

$$\left. \begin{array}{ll} \mathbf{w}^T \mathbf{x}_i + w_0 \geq 1 - \xi_i & \text{iff } y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + w_0 \leq -1 + \xi_i & \text{iff } y_i = -1 \end{array} \right\} y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i \geq 0$$

- ✓ Jaké bude naše kvadratické optimalizační kritérium?

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}, \xi^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N \xi_n$$

- ✓ Kolik omezení budeme mít?

QP pro neseparabilní data

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$ a minimalizovat součet vzdáleností špatně zaklasifikovaných trén. příkladů k jejich správnému umístění.

- ✓ Jak budou vypadat omezení? Zavedeme tzv. *slack variables* ξ_i :

$$\left. \begin{array}{ll} \mathbf{w}^T \mathbf{x}_i + w_0 \geq 1 - \xi_i & \text{iff } y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + w_0 \leq -1 + \xi_i & \text{iff } y_i = -1 \end{array} \right\} y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i \geq 0$$

- ✓ Jaké bude naše kvadratické optimalizační kritérium?

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}, \xi^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N \xi_n$$

- ✓ Kolik omezení budeme mít?

Stejně, jako je trénovacích bodů, tj. N .

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$ a minimalizovat součet vzdáleností špatně zaklasifikovaných trén. příkladů k jejich správnému umístění.

- ✓ Jak budou vypadat omezení? Zavedeme tzv. *slack variables* ξ_i :

$$\left. \begin{array}{ll} \mathbf{w}^T \mathbf{x}_i + w_0 \geq 1 - \xi_i & \text{iff } y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + w_0 \leq -1 + \xi_i & \text{iff } y_i = -1 \end{array} \right\} y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i \geq 0$$

- ✓ Jaké bude naše kvadratické optimalizační kritérium?

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}, \xi^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N \xi_n$$

- ✓ Kolik omezení budeme mít?

Stejně, jako je trénovacích bodů, tj. N .

V této formulaci jsou chyby. Dokážete je najít?

QP pro neseparabilní data

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$ a minimalizovat součet vzdáleností špatně zaklasifikovaných trén. příkladů k jejich správnému umístění.

- ✓ Jak budou vypadat omezení? Zavedeme tzv. *slack variables* ξ_i :

$$\left. \begin{array}{ll} \mathbf{w}^T \mathbf{x}_i + w_0 \geq 1 - \xi_i & \text{iff } y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + w_0 \leq -1 + \xi_i & \text{iff } y_i = -1 \end{array} \right\} y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i \geq 0$$

$$\xi_i \geq 0 \quad \forall i, i = 1, \dots, N$$

- ✓ Jaké bude naše kvadratické optimalizační kritérium?

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}, \xi^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N \xi_n$$

- ✓ Kolik omezení budeme mít?

Stejně, jako je trénovacích bodů, tj. N .

QP pro neseparabilní data

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$ a minimalizovat součet vzdáleností špatně zaklasifikovaných trén. příkladů k jejich správnému umístění.

- ✓ Jak budou vypadat omezení? Zavedeme tzv. *slack variables* ξ_i :

$$\left. \begin{array}{ll} \mathbf{w}^T \mathbf{x}_i + w_0 \geq 1 - \xi_i & \text{iff } y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + w_0 \leq -1 + \xi_i & \text{iff } y_i = -1 \end{array} \right\} y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i \geq 0$$

$$\xi_i \geq 0 \quad \forall i, i = 1, \dots, N$$

- ✓ Jaké bude naše kvadratické optimalizační kritérium?

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}, \xi^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N \xi_n$$

- ✓ Kolik omezení budeme mít?

Dvojnásobek počtu trénovacích bodů, tj. $2N$.

QP pro neseparabilní data

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v

Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

Chceme maximalizovat velikost marginu $M = \frac{2}{|w|}$ a minimalizovat součet vzdáleností špatně zaklasifikovaných trén. příkladů k jejich správnému umístění.

- ✓ Jak budou vypadat omezení? Zavedeme tzv. *slack variables* ξ_i :

$$\left. \begin{array}{ll} \mathbf{w}^T \mathbf{x}_i + w_0 \geq 1 - \xi_i & \text{iff } y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + w_0 \leq -1 + \xi_i & \text{iff } y_i = -1 \end{array} \right\} y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i \geq 0$$

$$\xi_i \geq 0 \quad \forall i, i = 1, \dots, N$$

- ✓ Jaké bude naše kvadratické optimalizační kritérium?

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}, \xi^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N \xi_n$$

- ✓ Kolik omezení budeme mít?

Dvojnásobek počtu trénovacích bodů, tj. $2N$.

- ✓ Původní úloha QP měla $D + 1$ proměnných: $w_0, \dots, w_D$.

Nová úloha QP má $D + 1 + N$ proměnných: $w_0, \dots, w_D, \xi_1, \dots, \xi_N$.

Primární úloha QP:

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}, \tilde{\xi}^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0, \tilde{\xi}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N \tilde{\xi}_n$$

s omezeními $\forall i, i = 1, \dots, N$

$$\begin{aligned} y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \tilde{\xi}_i &\geq 0 \\ \tilde{\xi}_i &\geq 0 \end{aligned}$$

Primární úloha QP:

s omezeními $\forall i, i = 1, \dots, N$

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}, \xi^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N \xi_n$$

$$\begin{aligned} y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i &\geq 0 \\ \xi_i &\geq 0 \end{aligned}$$

Metoda Lagrangových multiplikátorů nahrazuje hledání stacionárních bodů funkce D proměnných s K omezeními hledáním stac. bodů neomezené funkce $D + K$ proměnných: zavádí nové proměnné, tzv. *Lagrangeovy multiplikátory*, pro každé omezení a definuje novou funkci, tzv. *Lagrangián*, sestávající z původní funkce, omezení a multiplikátorů.

Primární úloha QP:

s omezeními $\forall i, i = 1, \dots, N$

$$(\mathbf{w}^{\text{OPT}}, w_0^{\text{OPT}}, \xi^{\text{OPT}}) = \arg \min_{\mathbf{w}, w_0, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^N \xi_n$$

$$\begin{aligned} y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i &\geq 0 \\ \xi_i &\geq 0 \end{aligned}$$

Metoda Lagrangových multiplikátorů nahrazuje hledání stacionárních bodů funkce D proměnných s K omezeními hledáním stac. bodů neomezené funkce $D + K$ proměnných: zavádí nové proměnné, tzv. *Lagrangeovy multiplikátory*, pro každé omezení a definuje novou funkci, tzv. *Lagrangián*, sestávající z původní funkce, omezení a multiplikátorů.

$$L(\mathbf{w}, w_0, \xi_i, \alpha_i, \mu_i) = \frac{1}{2} |\mathbf{w}|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i \{y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i\} - \sum_{i=1}^N \mu_i \xi_i \quad (2)$$

kde

- ✓ $\alpha_i \geq 0$ jsou Lagr. multiplikátory pro podmínky správné klasifikace bodů, a
- ✓ $\mu_i \geq 0$ jsou Lagr. multiplikátory pro podmínky kladnosti ξ_i .

Lagrangián je třeba minimalizovat vzhledem k *primárním proměnným* $\mathbf{w}$, w_0 a ξ_i a maximalizovat vzhledem k *duálním proměnným* α_i a μ_i .

Duální úlohu QP dostaneme, pokud z Lagrangiánu

$$L(\mathbf{w}, w_0, \xi_i, \alpha_i, \mu_i) = \frac{1}{2}|\mathbf{w}|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i \{y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i\} - \sum_{i=1}^N \mu_i \xi_i$$

vyloučíme primární proměnné $\mathbf{w}$, w_0 a ξ_i .

Duální úlohu QP dostaneme, pokud z Lagrangiánu

$$L(\mathbf{w}, w_0, \xi_i, \alpha_i, \mu_i) = \frac{1}{2}|\mathbf{w}|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i \{y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i\} - \sum_{i=1}^N \mu_i \xi_i$$

vyloučíme primární proměnné $\mathbf{w}$, w_0 a ξ_i .

Pro stacionární bod platí:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^N \alpha_i y_i = 0 \quad \Longrightarrow \quad \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L}{\partial w_0} = - \sum_{i=1}^N \alpha_i y_i = 0 \quad \Longrightarrow \quad \sum_{i=1}^N \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \quad \Longrightarrow \quad C = \alpha_i + \mu_i$$

Duální úlohu QP dostaneme, pokud z Lagrangiánu

$$L(\mathbf{w}, w_0, \xi_i, \alpha_i, \mu_i) = \frac{1}{2}|\mathbf{w}|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i \{y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i\} - \sum_{i=1}^N \mu_i \xi_i$$

vyloučíme primární proměnné $\mathbf{w}$, w_0 a ξ_i .

Pro stacionární bod platí:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^N \alpha_i y_i = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L}{\partial w_0} = - \sum_{i=1}^N \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^N \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \quad \Rightarrow \quad C = \alpha_i + \mu_i$$

Po dosazení zpět do L :

$$L_D = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^N \alpha_i \xi_i + \sum_{i=1}^N \mu_i \xi_i - \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^N \alpha_i y_i w_0 + \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \alpha_i \xi_i - \sum_{i=1}^N \mu_i \xi_i$$

Duální úlohu QP dostaneme, pokud z Lagrangiánu

$$L(\mathbf{w}, w_0, \xi_i, \alpha_i, \mu_i) = \frac{1}{2}|\mathbf{w}|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i \{y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i\} - \sum_{i=1}^N \mu_i \xi_i$$

vyloučíme primární proměnné $\mathbf{w}$, w_0 a ξ_i .

Pro stacionární bod platí:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^N \alpha_i y_i = 0 \quad \implies \quad \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L}{\partial w_0} = - \sum_{i=1}^N \alpha_i y_i = 0 \quad \implies \quad \sum_{i=1}^N \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \quad \implies \quad C = \alpha_i + \mu_i$$

Po dosazení zpět do L a zjednodušení:

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \tag{3}$$

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v
Langrangeově funkci
Proč duální formulace?

Kernel trik

Závěr

- ✓ Optimalizační kritérium:

$$(\alpha^{\text{OPT}}) = \arg \max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

- ✓ S následujícími omezeními:

$$0 \leq \alpha_i \leq C \quad \forall i = 1, \dots, N$$

$$\sum_{i=1}^N \alpha_i y_i = 0$$

- ✓ Pak můžeme vyjádřit:

$$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \quad (4)$$

$$w_0 = y_K - \mathbf{w}^T \mathbf{x}_K, \quad (5)$$

kde $\mathbf{x}_K$ je jakýkoli bod s $0 < \alpha_k < C$ ($\mathbf{x}_K$ je pak support vector ležící v plus- nebo minus-rovině a $\zeta_K = 0$).

- ✓ Klasifikovat nové body pak lze pomocí funkce

$$f(\mathbf{x}, \mathbf{w}, w_0) = \text{sign}(\mathbf{w}^T \mathbf{x} + w_0) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + w_0\right) \quad (6)$$

- ✓ Data s $\alpha_k > 0$ jsou support vectors, takže tento součet může být jen přes support vectors.

Lagrangian

$$L(\mathbf{w}, w_0, \xi_i, \alpha_i, \mu_i) = \frac{1}{2}|\mathbf{w}|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i \{y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i\} - \sum_{i=1}^N \mu_i \xi_i$$

je třeba minimalizovat vzhledem k *primárním proměnným* $\mathbf{w}$, w_0 a ξ_i a maximalizovat vzhledem k *duálním proměnným* α_i a μ_i .

1. Pokud bod $\mathbf{x}_i$ leží na nesprávné straně plus- nebo minus-roviny:

- ✓ $y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 < 0$, takže $\xi_i > 0$ tak, aby $y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i = 0$
- ✓ $\xi_i > 0$ a L je třeba maximalizovat vzhledem k μ_i , takže μ_i musí být co nejmenší, tj. $\mu_i = 0$
- ✓ $C = \alpha_i + \mu_i$ a $\mu_i = 0$, takže $\alpha_i = C$

2. Pokud bod $\mathbf{x}_i$ leží na správné straně plus- nebo minus-roviny:

- ✓ $y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 > 0$, takže $\xi_i = 0$
- ✓ $y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 + \xi_i > 0$ a L je třeba max. vzhledem k α_i , takže α_i musí být co nejmenší, tj. $\alpha_i = 0$
- ✓ $C = \alpha_i + \mu_i$ a $\alpha_i = 0$, takže $\mu_i = C$

3. Pokud bod $\mathbf{x}_i$ leží přímo v plus- nebo minus-rovině:

- ✓ $y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1 = 0$, takže $\xi_i = 0$
- ✓ $0 < \mu_i < C$
- ✓ $0 < \alpha_i < C$

Proč duální formulace?

Optimální rozdělující
nadplocha

Kvadratické
programování

Učení pomocí
kvadratického
programování

QP pro maximální
margin

Lineárně
neseparovatelná data

QP pro neseparabilní
data

Lagrangeova funkce

Duální úloha QP

QP pro duální úlohu

Vztahy proměnných v
Langrangeově funkci

Proč duální formulace?

Kernel trik

Závěr

- ✓ Úloha QP v duální formě je pro QP solvery snáze řešitelná než QP v primární formě.
- ✓ Duální úloha umožňuje využít dalších triků, o kterých se teprve dozvíte.

Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Opakování: rozšíření
báze

Kvadratické bazové
funkce

Kvadratické bazové
funkce (pokr.)

Kernel trik

Optimální rozdělující
nadplocha, rozšíření
báze, kernel trik

Support vector machine

Používané kernely

Demo: SVM s lineárním
jádre

Demo: SVM s
Gaussovským (RBF)
jádre

Závěr

Kernel trik

Opakování: rozšíření báze

Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Opakování: rozšíření
báze

Kvadratické bazové
funkce

Kvadratické bazové
funkce (pokr.)

Kernel trik

Optimální rozdělující
nadplocha, rozšíření
báze, kernel trik

Support vector machine

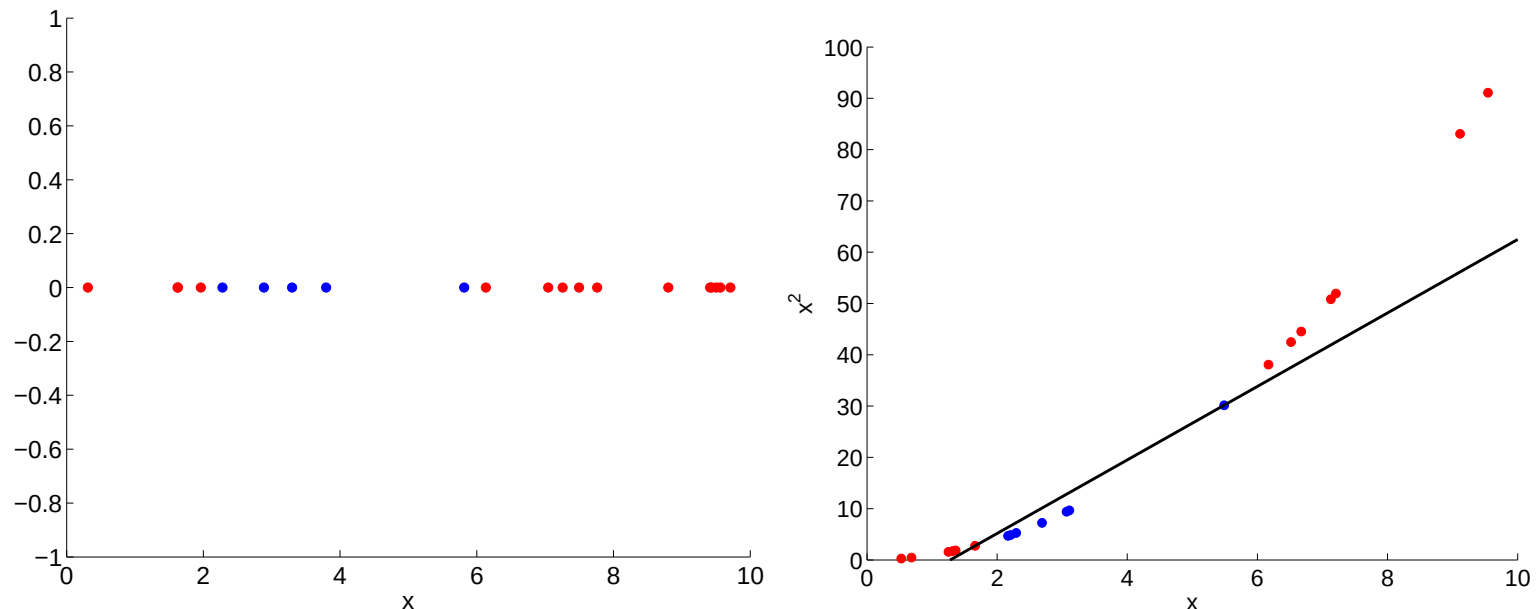
Používané kernely

Demo: SVM s lineárním
jádre

Demo: SVM s
Gaussovským (RBF)
jádre

Závěr

Zavedení nelineárních bazových funkcí umožňuje lineárním metodám vytvářet nelineární klasifikátory:



- ✓ použijme je také pro klasifikátor s maximálním marginem a
- ✓ budeme vytvářet nelineární „optimální“ klasifikátory.

Opakování: rozšíření báze

Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Opakování: rozšíření
báze

Kvadratické bazové
funkce

Kvadratické bazové
funkce (pokr.)

Kernel trik

Optimální rozdělující
nadplocha, rozšíření
báze, kernel trik

Support vector machine

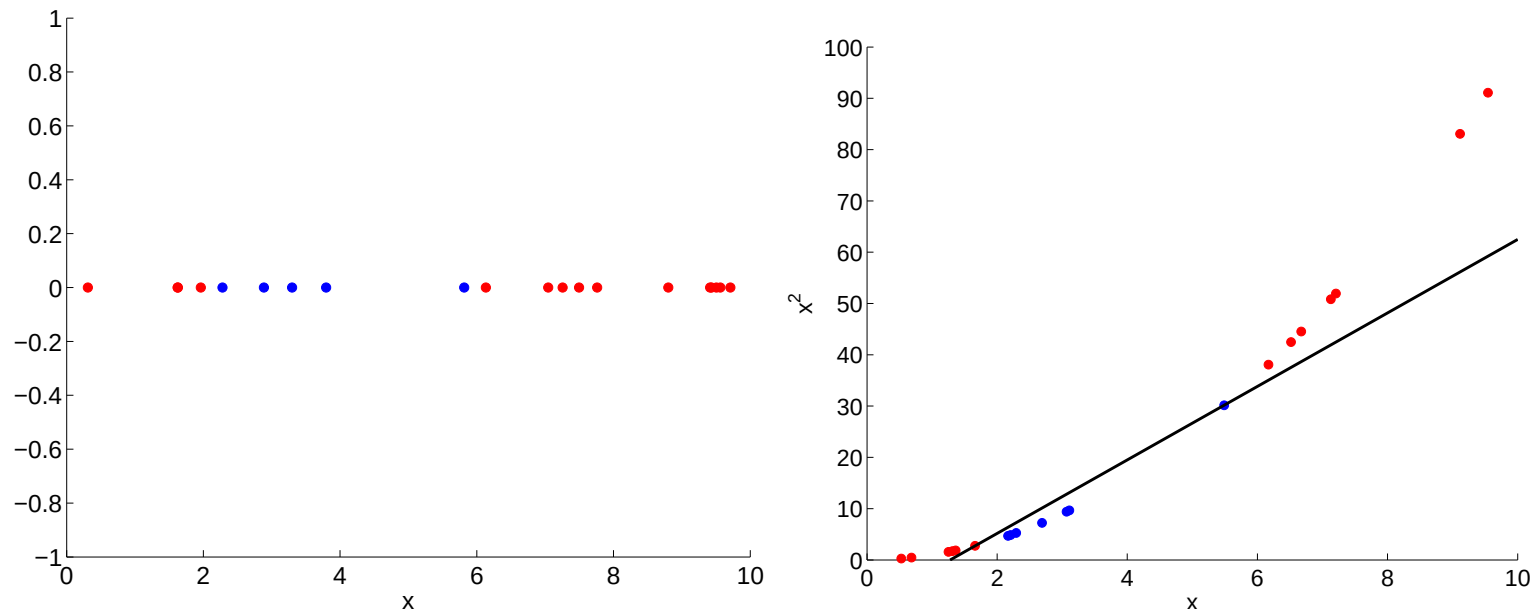
Používané kernely

Demo: SVM s lineárním
jádre

Demo: SVM s
Gaussovským (RBF)
jádre

Závěr

Zavedení nelineárních bazových funkcí umožňuje lineárním metodám vytvářet nelineární klasifikátory:



- ✓ použijme je také pro klasifikátor s maximálním marginem a
- ✓ budeme vytvářet nelineární „optimální“ klasifikátory.
- ✓ Q: „A to je celé?“

Opakování: rozšíření báze

Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Opakování: rozšíření
báze

Kvadratické bazové
funkce

Kvadratické bazové
funkce (pokr.)

Kernel trik

Optimální rozdělující
nadplocha, rozšíření
báze, kernel trik

Support vector machine

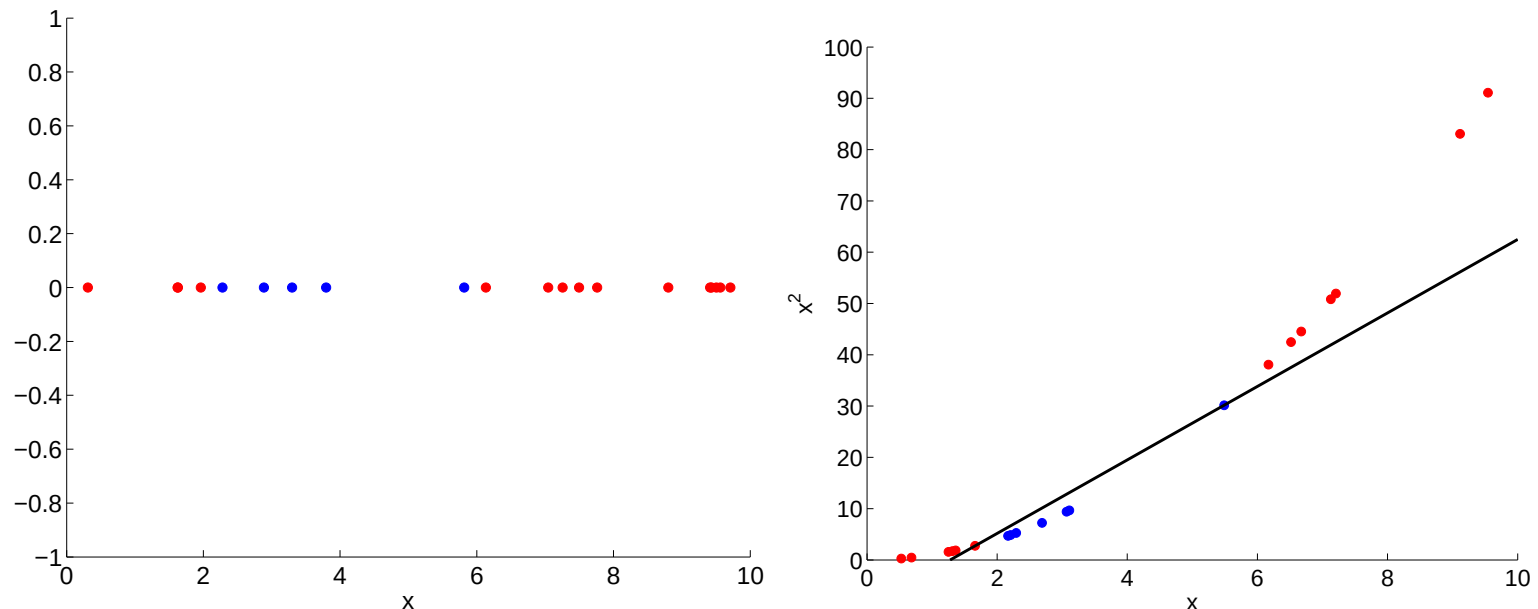
Používané kernely

Demo: SVM s lineárním
jádre

Demo: SVM s
Gaussovským (RBF)
jádre

Závěr

Zavedení nelineárních bazových funkcí umožňuje lineárním metodám vytvářet nelineární klasifikátory:



- ✓ použijme je také pro klasifikátor s maximálním marginem a
- ✓ budeme vytvářet nelineární „optimální“ klasifikátory.
- ✓ Q: „A to je celé?“
- ✓ A: „NIKOLIV! Je tu ještě jeden trik!“

Mapování do prostoru obrazů:

$$\phi_Q(\mathbf{x}) = \underbrace{(x_1^2, \sqrt{2}x_1x_2, \dots, \sqrt{2}x_1x_D, x_2^2, \dots, \sqrt{2}x_2x_D, \dots, x_D^2)}_{\frac{D(D+1)}{2} \text{ kvadratických členů}}, \underbrace{(\sqrt{2}x_1, \sqrt{2}x_2, \dots, \sqrt{2}x_D)}_{D \text{ lineárních členů}}, \underbrace{(1)}_{1 \text{ konst. člen}})^T \quad (7)$$

Dimenze prostoru obrazů:

$$D_\phi = \frac{D(D+1)}{2} + D + 1 = \frac{D^2 + 3D + 2}{2} = \frac{D(D+3)}{2} + 1 = \frac{(D+2)(D+1)}{2} = \binom{D+2}{2} = \mathcal{O}(D^2)$$

Duální úloha QP s bázovými funkcemi (všechny výskyty $\mathbf{x}$ nahradíme $\phi(\mathbf{x})$):

$$(\alpha^{\text{OPT}}) = \arg \max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

Pak můžeme vyjádřit:

$$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x}_i) \quad (8)$$

(9)

Klasifikovat nové body pak lze pomocí funkce

$$f(\mathbf{x}, \mathbf{w}, w_0) = \text{sign}(\mathbf{w}^T \phi(\mathbf{x}) + w_0) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + w_0\right) \quad (10)$$

Skalární součin bodů v obrazovém prostoru:

$$\phi(\mathbf{a})^T \phi(\mathbf{b}) = \sum_{i=1}^D a_i^2 b_i^2 + 2 \sum_{i=1}^D \sum_{j=i+1}^D a_i a_j b_i b_j + 2 \sum_{i=1}^D a_i b_i + 1 \quad (11)$$

Na jeho výpočet potřebujeme $\mathcal{O}(D^2)$ násobení a sčítání.

Podívejme se, z čisté zvědavosti, na jinou funkci vektorových argumentů $\mathbf{a}$ a $\mathbf{b}$:

$$\begin{aligned} K(\mathbf{a}, \mathbf{b}) &= (\mathbf{a}^T \mathbf{b} + 1)^2 = \\ &= (\mathbf{a}^T \mathbf{b})^2 + 2\mathbf{a}^T \mathbf{b} + 1 = \left(\sum_{i=1}^D a_i b_i\right)^2 + 2 \sum_{i=1}^D a_i b_i + 1 = \\ &= \sum_{i=1}^D \sum_{j=1}^D a_i b_i a_j b_j + 2 \sum_{i=1}^D a_i b_i + 1 = \\ &= \sum_{i=1}^D a_i^2 b_i^2 + 2 \sum_{i=1}^D \sum_{j=i+1}^D a_i a_j b_i b_j + 2 \sum_{i=1}^D a_i b_i + 1 \\ K(\mathbf{a}, \mathbf{b}) &= (\mathbf{a}^T \mathbf{b} + 1)^2 = \phi(\mathbf{a})^T \phi(\mathbf{b}) \end{aligned}$$

Na výpočet hodnoty funkce $K(\mathbf{a}, \mathbf{b})$ potřebujeme pouze $\mathcal{O}(D)$ násobení a sčítání.

Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Opakování: rozšíření
báze

Kvadratické bazové
funkce

Kvadratické bazové
funkce (pokr.)

Kernel trik

Optimální rozdělující
nadplocha, rozšíření
báze, kernel trik

Support vector machine

Používané kernely

Demo: SVM s lineárním
jádre

Demo: SVM s
Gaussovským (RBF)
jádre

Závěr

*Existují funkce dvou vektorových argumentů ($K(\mathbf{a}, \mathbf{b})$), jejichž hodnoty jsou rovny skalárnímu součinu ($\phi(\mathbf{a})^T \phi(\mathbf{b})$) vektorů vzniklých namapováním obou původních vektorů do mnohazměrného prostoru obrazů. Takovým funkcím se říká **jádrové funkce** nebo **kernely**.*

Kernel trik: Mějme lineární algoritmus, v němž se trénovací příklady vyskytují pouze ve skalárních součinech.

- ✓ Takový algoritmus lze znelinearizovat, když nahradíme skalární součiny dvou trénovacích vektorů jádrovou funkcí těchto vektorů.
- ✓ Výsledek je stejný, jako by se tentýž algoritmus učil v mnohazměrném prostoru nelineárních bazových funkcí.
- ✓ Díky kernelům ovšem není nutné vlastní mapování provádět, algoritmus je mnohem efektivnější.

QP pro duální úlohu s jádrovými funkcemi (všechny výskyty $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ nahradíme $K(\mathbf{x}_i, \mathbf{x}_j)$):

$$(\alpha^{\text{OPT}}) = \arg \max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

Klasifikovat nové body pak lze pomocí funkce

$$f(\mathbf{x}, \mathbf{w}, w_0) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + w_0\right) \quad (12)$$

Optimální rozdělující nadplocha, rozšíření báze, kernel trik

Optimální rozdělující nadplocha

Kvadratické programování

Kernel trik

Opakování: rozšíření báze

Kvadratické bazové funkce

Kvadratické bazové funkce (pokr.)

Kernel trik

Optimální rozdělující nadplocha, rozšíření báze, kernel trik

Support vector machine

Používané kernely

Demo: SVM s lineárním jádrem

Demo: SVM s Gaussovským (RBF) jádrem

Závěr

Připomeňme: při učení optimální rozdělující nadplochy se vektory $\mathbf{x}$ vyskytují pouze

v optimalizačním kritériu $\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$

a v diskriminační funkci $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^N \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + w_0)$.

Optimální rozdělující nadplocha, rozšíření báze, kernel trik

Optimální rozdělující nadplocha

Kvadratické programování

Kernel trik

Opakování: rozšíření báze

Kvadratické bazové funkce

Kvadratické bazové funkce (pokr.)

Kernel trik

Optimální rozdělující nadplocha, rozšíření báze, kernel trik

Support vector machine

Používané kernely

Demo: SVM s lineárním jádrem

Demo: SVM s Gaussovským (RBF) jádrem

Závěr

Připomeňme: při učení optimální rozdělující nadplochy se vektory $\mathbf{x}$ vyskytují pouze

v optimalizačním kritériu
$$\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

a v diskriminační funkci
$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + w_0\right).$$

Použijeme-li rozšíření báze, změní se

optimalizační kritérium na
$$\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$$

a diskriminační funkce na
$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) + w_0\right).$$

Optimální rozdělující nadplocha, rozšíření báze, kernel trik

Optimální rozdělující nadplocha

Kvadratické programování

Kernel trik

Opakování: rozšíření báze

Kvadratické bazové funkce

Kvadratické bazové funkce (pokr.)

Kernel trik

Optimální rozdělující nadplocha, rozšíření báze, kernel trik

Support vector machine

Používané kernely

Demo: SVM s lineárním jádrem

Demo: SVM s Gaussovským (RBF) jádrem

Závěr

Připomeňme: při učení optimální rozdělující nadplochy se vektory $\mathbf{x}$ vyskytují pouze

$$\text{v optimalizačním kritériu } \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{a v diskriminační funkci } f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + w_0\right).$$

Použijeme-li rozšíření báze, změní se

$$\text{optimalizační kritérium na } \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$$

$$\text{a diskriminační funkce na } f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) + w_0\right).$$

Místo skalárního součinu v obrazovém prostoru lze použít funkci $K(\mathbf{x}_i, \mathbf{x}_j)$

$$\text{v optimalizačním kritériu } \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{a v diskriminační funkci } f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + w_0\right).$$

Support vector machine

Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Opakování: rozšíření
báze

Kvadratické bazové
funkce

Kvadratické bazové
funkce (pokr.)

Kernel trik

Optimální rozdělující
nadplocha, rozšíření
báze, kernel trik

Support vector machine

Používané kernely

Demo: SVM s lineárním
jádre

Demo: SVM s
Gaussovským (RBF)
jádre

Závěr

Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Opakování: rozšíření
báze

Kvadratické bazové
funkce

Kvadratické bazové
funkce (pokr.)

Kernel trik

Optimální rozdělující
nadplocha, rozšíření
báze, kernel trik

Support vector machine

Používané kernely

Demo: SVM s lineárním
jádre

Demo: SVM s
Gaussovským (RBF)
jádre

Závěr

Support vector machine (SVM)

=

optimální rozdělující nadplocha

+

kernel trik

Používané kernely

Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Opakování: rozšíření
báze

Kvadratické bazové
funkce

Kvadratické bazové
funkce (pokr.)

Kernel trik

Optimální rozdělující
nadplocha, rozšíření
báze, kernel trik

Support vector machine

Používané kernely

Demo: SVM s lineárním
jádre

Demo: SVM s
Gaussovským (RBF)
jádre

Závěr

Polynomiální kernel:

$$K(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^T \mathbf{b} + 1)^d, \text{ kde } d \text{ je stupeň polynomu.} \quad (13)$$

Gaussovský (RBF) kernel:

$$K(\mathbf{a}, \mathbf{b}) = \exp\left(-\frac{|\mathbf{a} - \mathbf{b}|^2}{\sigma^2}\right), \text{ kde } \sigma^2 \text{ je „šířka“ Gaussiánu.} \quad (14)$$

Sigmoidální kernel:

$$K(\mathbf{a}, \mathbf{b}) = \tanh(\kappa \mathbf{a}^T \mathbf{b} + \delta) \quad (15)$$

Volba jádrové funkce a jejích parametrů je silně závislá na konkrétní řešené úloze.

Demo: SVM s lineárním jádrem

Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Opakování: rozšíření
báze

Kvadratické bazové
funkce

Kvadratické bazové
funkce (pokr.)

Kernel trik

Optimální rozdělující
nadplocha, rozšíření
báze, kernel trik

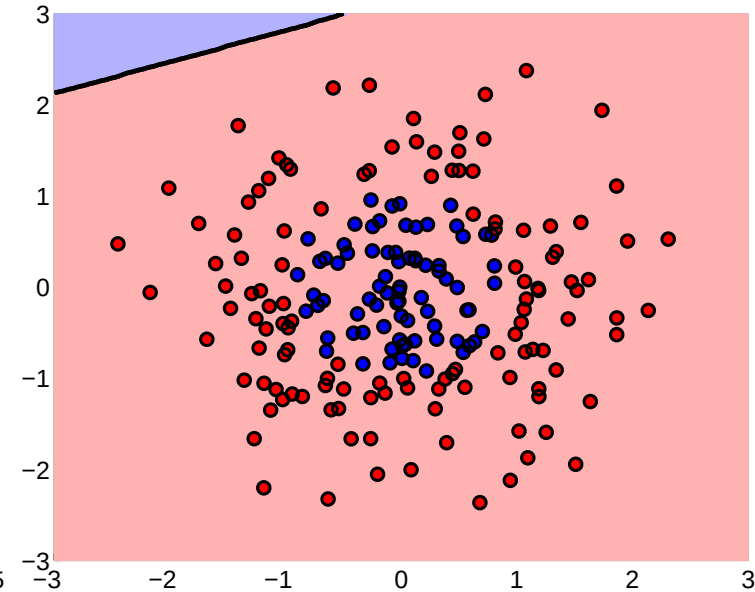
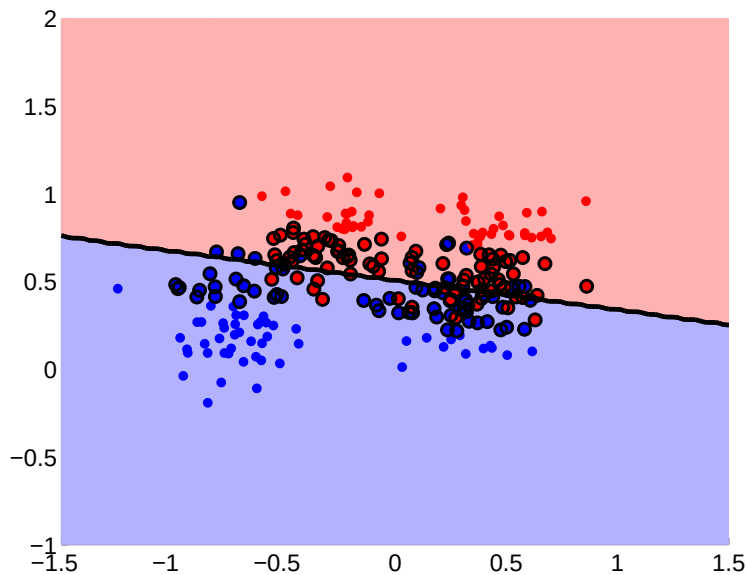
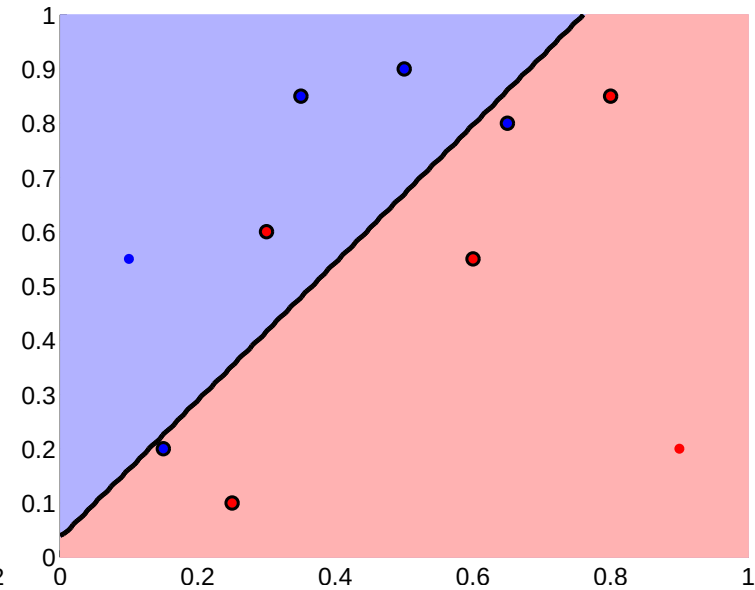
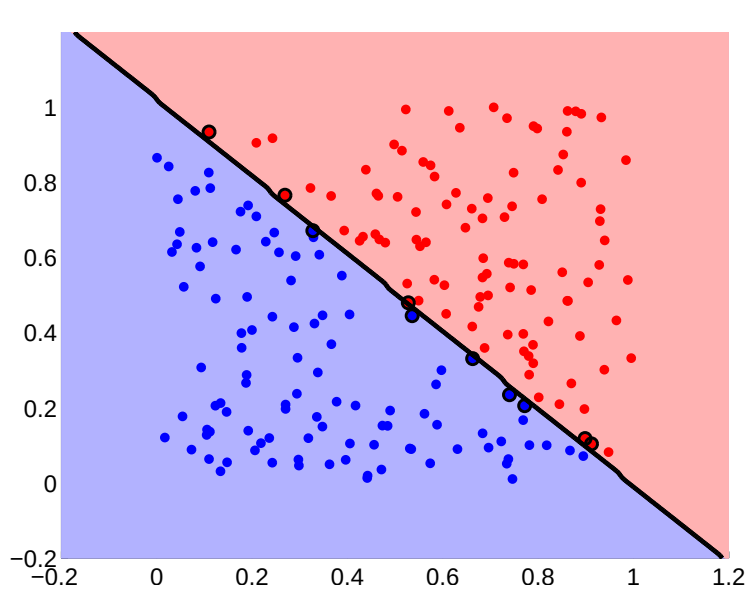
Support vector machine

Používané kernely

Demo: SVM s lineárním
jádre

Demo: SVM s
Gaussovským (RBF)
jádre

Závěr



Demo: SVM s Gaussovským (RBF) jádrem

Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Opakování: rozšíření
báze

Kvadratické bázevé
funkce

Kvadratické bázevé
funkce (pokr.)

Kernel trik

Optimální rozdělující
nadplocha, rozšíření
báze, kernel trik

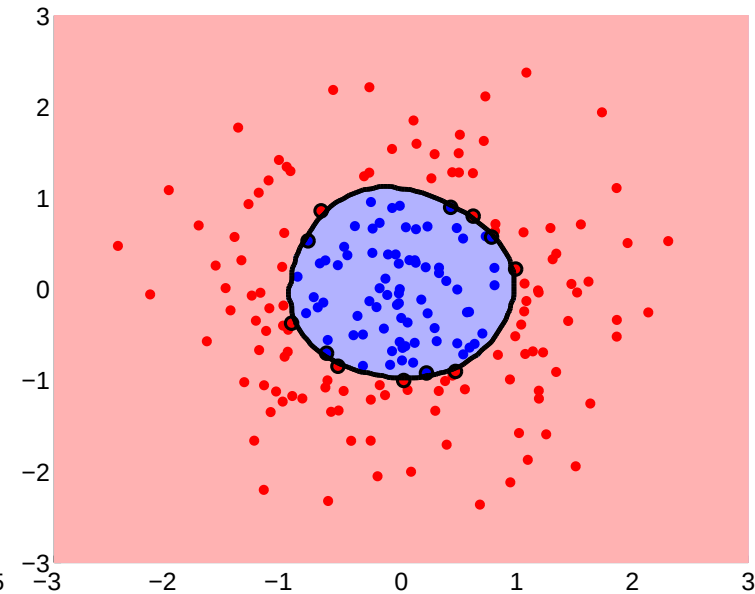
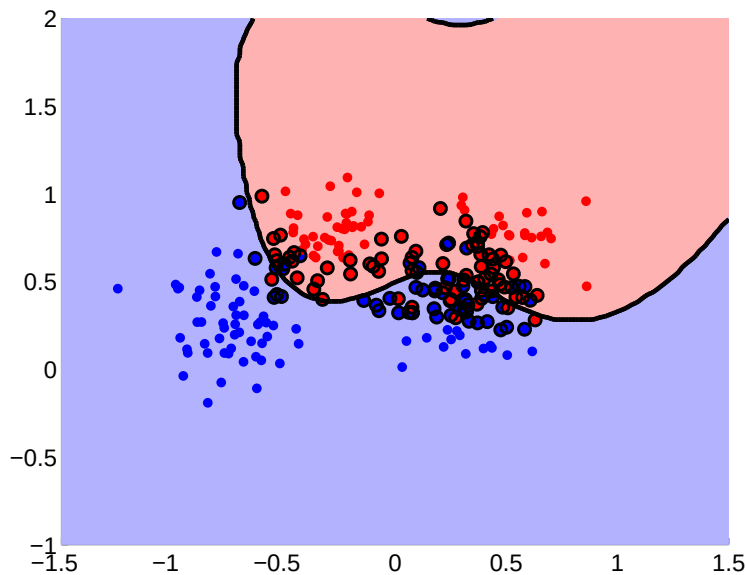
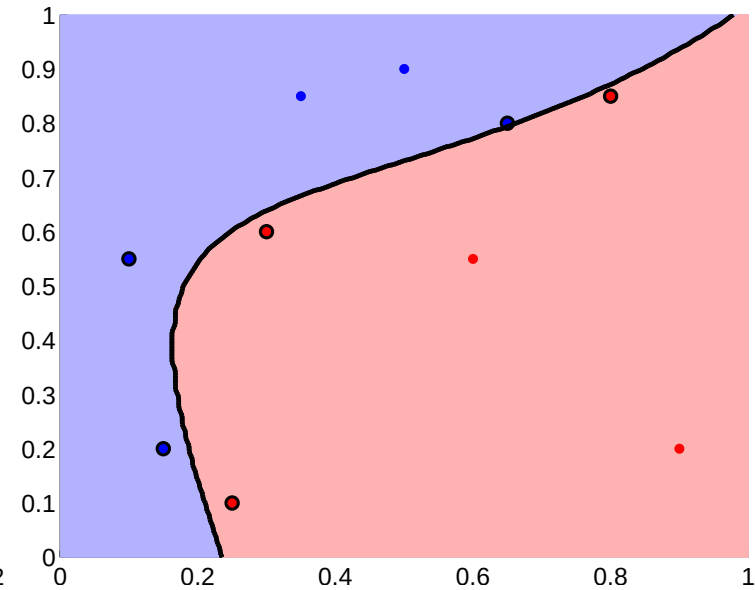
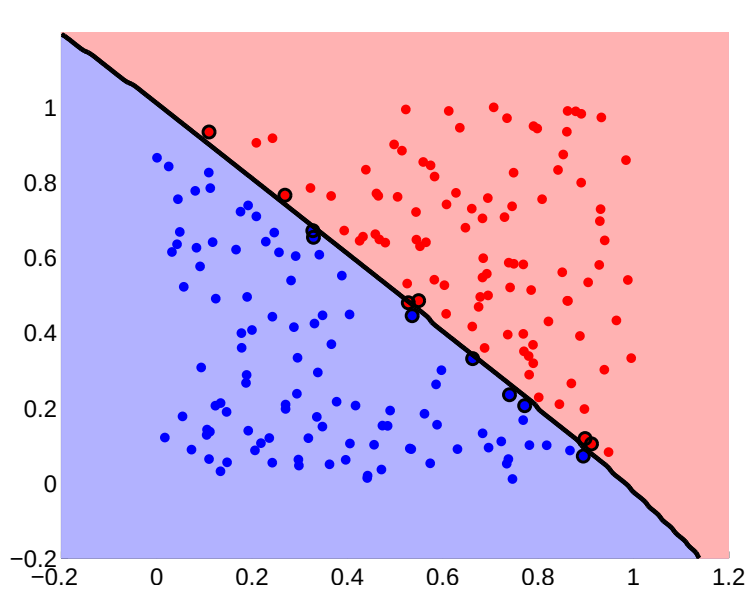
Support vector machine

Používané kernely

Demo: SVM s lineárním
jádre

Demo: SVM s
Gaussovským (RBF)
jádre

Závěr



Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Závěr

Shrnutí

Závěr

Optimální rozdělující
nadplocha

Kvadratické
programování

Kernel trik

Závěr

Shrnutí

- ✓ Maximalizace marginu $\implies$ optimální rozdělující nadplocha
- ✓ Maximalizaci marginu lze formulovat jako kvadratický program
- ✓ Příklad, kdy třídy nejsou lineárně separovatelné, lze ošetřit zavedením „opravných“ proměnných a minimalizací jejich velikosti
- ✓ Duální forma QP je jednodušší a příklady se v ní vyskytují pouze ve skalárních součinech
- ✓ Pomocí nelineárních báзовých funkcí lze vytvářet nelineární klasifikátory
- ✓ Efektivnější je to ovšem pomocí *kernel triku*, kdy skalární součiny nahradíme jádrovou funkcí