Uvod

SVM = Support Vector Machine = Metoda podpornih vektorjev

- Vapnik in Lerner 1963 (generalized portrait)
- jedra: Aronszajn 1950; Aizerman 1964; Wahba 1990, Poggio in Girosi 1990
- Boser, Guyon in Vapnik na COLT 1992 (osnovna oblika SVM, jedra)
- Cortes in Vapnik, *Machine Learning* 1995 (soft margin)
- poplava člankov po 1995

\+ dobro deluje v praksi

\+ fleksibilna (različna jedra $\rightarrow$ različni prostori hipotez)

\+ robustna (kolikor toliko odporna na pretirano prilagajanje;
ne moti je veliko število atributov)

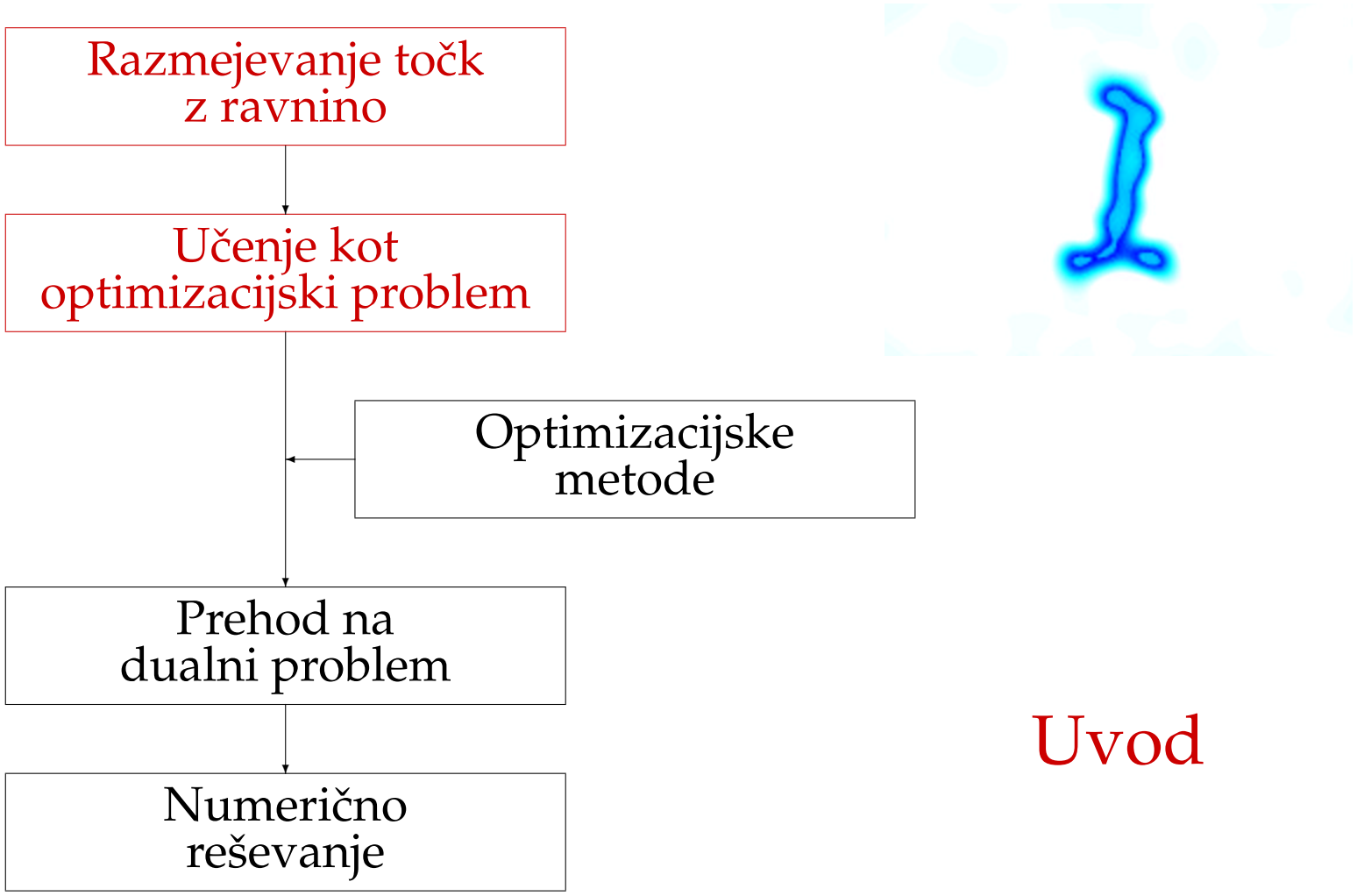
\+ dobro teoretično podprta

– malo težje jo je razumeti; kompleksnejša implementacija

– modeli niso preveč razumljivi

– časovno zahtevnejša

Kazalo


Razmejevanje točk
z ravnino
Učenje kot
optimizacijski problem
Optimizacijske
metode
Prehod na
dualni problem
Numerično
reševanje

Uvod

Motivacija

- Imamo populacijo *primerkov*, ki so predstavljeni z vektorji iz $\mathbb{R}^d$.
- Imamo dva *razreda*, *pozitivnega* in *negativnega*.
  Vsak primerek spada v enega od teh dveh razredov.
- Učna množica: pari $(\mathbf{x}_i, y_i)$ za $i \in 1..l$.
  $\mathbf{x}_i \in \mathbb{R}^d$ je vektor, $y_i \in \{1, -1\}$ je njegova oznaka razreda.
- Radi bi dobili *klasifikator* (model, hipotezo), ki bi razločeval ta dva razreda.

Za začetek se omejimo na to, da bi oba razreda razmejili z neko (hiper)ravnino.

Enačba ravnine

- *Ravnino* prepoznamo po tem, da je cela pravokotna na neko smer. Vektor $\mathbf{w}$, pravokoten na ravnino, imenujemo *normala* ravnine.
- Potem je vsak vektor, ki leži v celoti znotraj ravnine, pravokoten na $\mathbf{w}$.
- Naj bo $\mathbf{x}_0$ poljubna točka z ravnine. Potem za vsako točko $\mathbf{x}$ z ravnine velja:

$$\mathbf{x} - \mathbf{x}_0 \perp \mathbf{w}$$

Dva vektorja, $\mathbf{x}$ in $\hat{\mathbf{u}}$, sta pravokotna natanko tedaj, ko je njun *skalarni produkt* enak 0.

$$(\mathbf{x}, \mathbf{u}) = \langle \mathbf{x}, \mathbf{u} \rangle = \langle \mathbf{x} \cdot \mathbf{u} \rangle = \mathbf{x}^T \mathbf{u} = \sum_{i=1}^{d} x_i u_i.$$

Torej

$$\begin{aligned} \mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) &= 0 \\ \mathbf{w}^T\mathbf{x} - \mathbf{w}^T\mathbf{x}_0 &= 0 \\ \mathbf{w}^T\mathbf{x} + b &= 0 \text{ (za } b = -\mathbf{w}^T\mathbf{x}_0) \\ \sum_{i=1}^{d} w_i x_i + b &= 0 \end{aligned}$$

Nad in pod ravnino

- Začnimo v ravnini, v neki točki $\mathbf{x}$, in se premaknimo v smeri $\mathbf{w}$. Za naš novi položaj $\hat{\mathbf{x}} = \mathbf{x} + \lambda\mathbf{w}$ velja:

$$\mathbf{w}^T\hat{\mathbf{x}} + b = \mathbf{w}^T(\mathbf{x} + \lambda\mathbf{w}) + b = \mathbf{w}^T\mathbf{x} + b + \lambda\mathbf{w}^T\mathbf{w} = \lambda\mathbf{w}^T\mathbf{w} > 0.$$

  Če bi se premikali v nasprotni smeri, bi za novi položaj veljalo

$$\mathbf{w}^T\hat{\mathbf{x}} + b < 0.$$

- Ravnina torej razdeli prostor na dva polprostora in za poljubno točko je zelo preprosto ugotoviti, v katerem leži.
- Za naš problem bi bilo torej ugodno, če bi ravnino postavili tako, da bi pozitivni primerki ležali na eni strani ravnine, negativni pa na drugi. Potem bi vzeli preprosto:

$$\mathit{napoved}(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}^T\mathbf{x} + b).$$

Razmejevanje učnih primerkov z ravnino

Pozitivni primerki naj ležijo na pozitivni strani, negativni pa na negativni:

$$y_i = +1 \implies \mathbf{w}^T\mathbf{x}_i + b \geq 0$$
$$y_i = -1 \implies \mathbf{w}^T\mathbf{x}_i + b \leq 0,$$

kar lahko združimo v:

$$\text{za vsak } i: \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 0.$$

Toda:

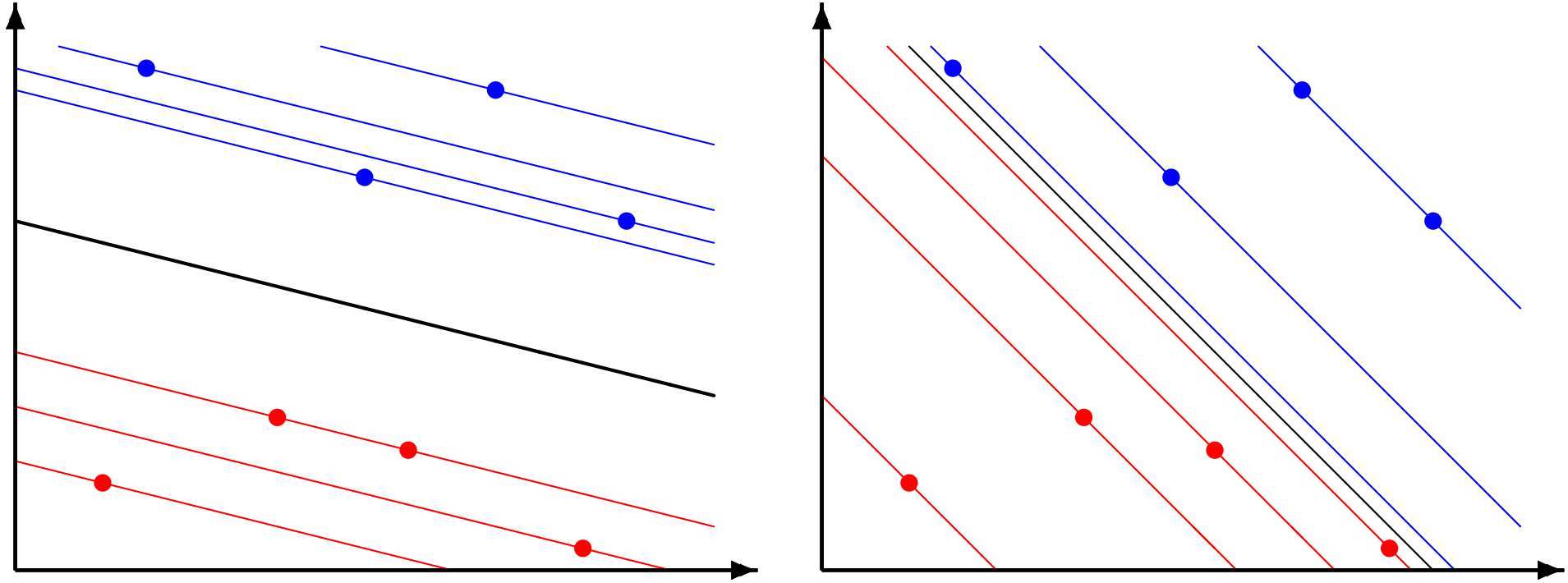
- Tej zahtevi mogoče ustreza veliko ravnin.
- Tej zahtevi mogoče ne ustreza nobena ravnina.

Za katero ravnino naj se odločimo?

Rob

Ko računamo $\mathbf{w}^T\mathbf{x} + b$ za razne $\mathbf{x}$, se pri vsaki točki pozna le to, kako daleč je od razmejitvene ravnine (in na kateri strani je).

- Torej si ne želimo, da bi bili učni primerki preblizu razmejitvene ravnine, ker postanemo potem bolj občutljivi na majhne spremembe in premike.



Prostor med ravnini najbližjima primerkoma na eni in na drugi strani imenujemo *rob* (*margin*) in si želimo, da bi bil čim širši.

Rob

Naj bo rob omejen z ravninama:

$$\mathbf{w}^T\mathbf{x} + b = -\gamma \qquad \text{in} \qquad \mathbf{w}^T\mathbf{x} + b = \gamma.$$

Učni primerki nam torej postavljajo takšne pogoje:

$$y_i(\mathbf{w}^T\mathbf{x} + b) \geq \gamma$$

Kako širok je ta rob?

- Postavimo se v neko točko $\mathbf{x}_1$ na prvi ravnini:

  $$\mathbf{w}^T\mathbf{x}_1 + b = -\gamma$$

  Premikajmo se v smeri normale, dokler ne naletimo na drugo ravnino:

  $$\mathbf{w}^T\mathbf{x}_2 + b = \gamma \qquad \text{za} \qquad \mathbf{x}_2 = \mathbf{x}_1 + \lambda\mathbf{w}.$$

  Torej je

  $$\begin{aligned} \mathbf{w}^T(\mathbf{x}_2 - \mathbf{x}_1) &= 2\gamma \\ \lambda\mathbf{w}^T\mathbf{w} &= 2\gamma \\ \lambda &= 2\gamma/||\mathbf{w}||^2 \end{aligned}$$

  Širina roba je $||\lambda\mathbf{w}|| = |\lambda| \cdot ||\mathbf{w}|| = \frac{2\gamma}{||\mathbf{w}||^2}||\mathbf{w}|| = 2\gamma/||\mathbf{w}||.$

Širina roba

Vidimo: rob je pri dani γ tem širši, čim krajša je normala $\mathbf{w}$.

Parameter γ je odveč:

- Če ravnina
$$\mathbf{w}^T\mathbf{x} + b = 0$$
ustreza pogojem
$$\forall i: \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq \gamma,$$
lahko isto ravnino opišemo kot
$$\hat{\mathbf{w}}^T\mathbf{x} + \hat{b}$$
za
$$\hat{\mathbf{w}} = \gamma^{-1}\mathbf{w}, \quad \hat{b} = b/\gamma$$
in bo ustrezala pogojem:
$$\forall i: \quad y_i(\hat{\mathbf{w}}^T\mathbf{x}_i + \hat{b}) \geq 1.$$

- Torej ni nič narobe, če vzamemo $\gamma = 1$.

Rob je potem širok $2/||\mathbf{w}||$ in če hočemo najširši rob, moramo zahtevati najkrajšo normalo $\mathbf{w}$.

Optimizacijski problem

Zdaj lahko zastavimo iskanje ravnine kot optimizacijski problem:

$$\begin{aligned} &\text{minimiziraj } \tfrac{1}{2}||\mathbf{w}||^2 \\ &\text{pri pogojih } \forall i \in 1..l : y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \end{aligned}$$

Faktor $\frac{1}{2}$ je tu le zaradi lepšega nadaljevanja izpeljave.

Kaj pa, če pozitivnih in negativnih primerkov ne moremo razmejiti z ravnino? Vpeljimo *kazenske spremenljivke* ξ_i in pogoj

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

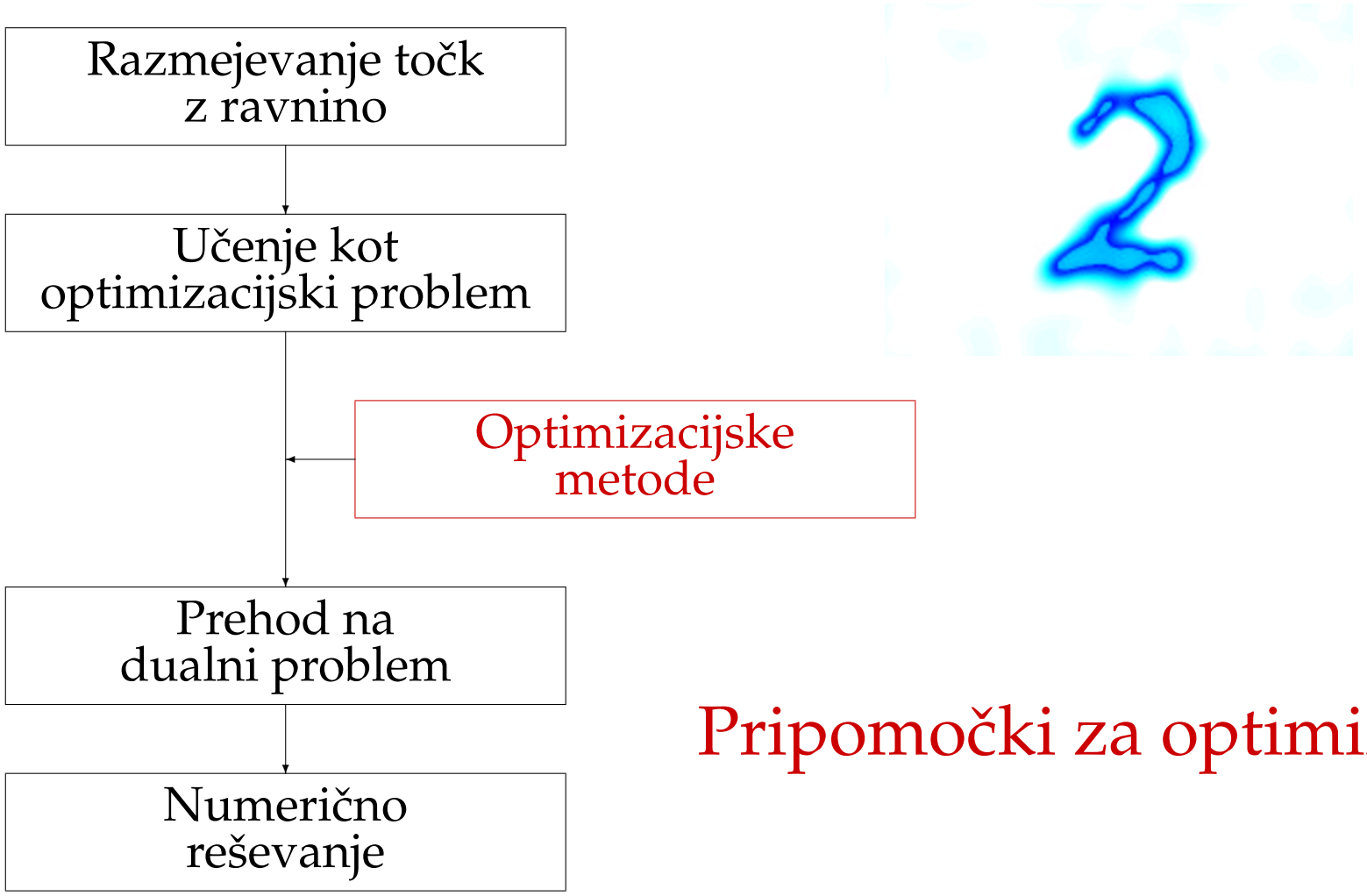
omilimo v

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{za neko} \quad \xi_i \geq 0.$$

Če gre s $\xi_i = 0$, toliko lepše, drugače pa jo upoštevajmo v kriterijski funkciji:

$$\begin{aligned} &\text{minimiziraj } \tfrac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l}\xi_i \\ &\text{pri pogojih } \forall i \in 1..l : y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned}$$

Zdaj potrebujemo kakšna matematična orodja, da se bomo lahko lotili tega problema.



Pripomočki za optimizacijo

Lagrangeovi multiplikatorji

Recimo, da imamo takšen optimizacijski problem:

$$\begin{array}{ll} \text{minimiziraj} & f(\mathbf{x}) \qquad (f \text{ imenujemo } \textit{kriterijska funkcija}) \\ \text{pri pogojih} & \mathbf{x} \in \mathbb{R}^d, \quad \forall i \in 1..l : f_i(\mathbf{x}) \geq 0 \end{array}$$

Če $\mathbf{x}$ ustreza vsem pogojem $f_i(\mathbf{x}) \geq 0$, pravimo, da je *dopustna rešitev*.
Množico vseh dopustnih rešitev označimo s Φ.

Nasvet: vpeljimo *Lagrangeove multiplikatorje* $\mathbf{u} = (u_1, \ldots, u_l)$ (ki bodo vedno nenegativna realna števila) in definirajmo Lagrangeovo funkcijo

$$L(\mathbf{x}, \mathbf{u}) := f(\mathbf{x}) - \sum_{i=1}^{l} u_i f_i(\mathbf{x}).$$

Izpeljimo iz nje dve novi funkciji: $\hat{f}(\mathbf{x}) := \max\limits_{\mathbf{u} \in (\mathbb{R}_0^+)^l} L(\mathbf{x}, \mathbf{u})$ in $g(\mathbf{u}) := \min\limits_{\mathbf{x} \in \mathbb{R}^d} L(\mathbf{x}, \mathbf{u})$.

Hitro se vidi:
$$\hat{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{če } \mathbf{x} \in \Phi \\ +\infty, & \text{če } \mathbf{x} \notin \Phi \end{cases}$$

In $\forall \mathbf{x}, \mathbf{u}$:
$$g(\mathbf{u}) = \min_{\tilde{\mathbf{x}} \in \mathbb{R}^d} L(\tilde{\mathbf{x}}, \mathbf{u}) \leq L(\mathbf{x}, \mathbf{u}) \leq \max_{\tilde{\mathbf{u}} \in (\mathbb{R}_0^+)^l} L(\mathbf{x}, \tilde{\mathbf{u}}) = f(\mathbf{x}).$$

Torej tudi:
$$\max_{\mathbf{u} \in (\mathbb{R}_0^+)^l} g(\mathbf{u}) \leq \min_{\mathbf{x} \in \mathbb{R}^d} \hat{f}(\mathbf{x}).$$

Sedla

$L(\mathbf{x}, \mathbf{u})$ ima v točki $(\mathbf{x}^s, \mathbf{u}^s)$ *sedlo* natanko tedaj, ko velja:

$$\forall \mathbf{x} : L(\mathbf{x}, \mathbf{u}^s) \geq L(\mathbf{x}^s, \mathbf{u}^s) \qquad \text{in} \qquad \forall \mathbf{u} : L(\mathbf{x}^s, \mathbf{u}) \leq L(\mathbf{x}^s, \mathbf{u}^s).$$

V $(\mathbf{x}^s, \mathbf{u}^s)$ ima torej L minimum po $\mathbf{x}$ in maksimum po $\mathbf{u}$. Zato je

$$\hat{f}(\mathbf{x}^s) = L(\mathbf{x}^s, \mathbf{u}^s) = g(\mathbf{u}^s).$$

Na prejšnji strani smo videli, da je $\forall \mathbf{x}, \mathbf{u} : \hat{f}(\mathbf{x}) \geq g(\mathbf{u})$,
torej tudi za $\mathbf{u} = \mathbf{u}^s$: $\forall \mathbf{x} : \hat{f}(\mathbf{x}) \geq g(\mathbf{u}^s) = \hat{f}(\mathbf{x}^s)$.
Zato doseže $\hat{f}(\mathbf{x})$ v $\mathbf{x} = \mathbf{x}^s$ svoj minimum.

Podobno mora pri $\mathbf{x} = \mathbf{x}^s$ veljati $\forall \mathbf{u} : g(\mathbf{u}) \leq \hat{f}(\mathbf{x}^s) = g(\mathbf{u}^s)$,
zato doseže g v $\mathbf{u} = \mathbf{u}^s$ svoj maksimum.

Tedaj je torej $\hat{f}(\mathbf{x}^s) = \min_{\mathbf{x}} \hat{f}(\mathbf{x}) = \max_{\mathbf{u}} g(\mathbf{u}) = g(\mathbf{u}^s)$.

Slaterjev *zadostni pogoj* za obstoj sedla:

- f in f_i morajo biti vse konveksne in
- obstajati mora $\mathbf{x}$, za katerega je $\forall i : f_i(\mathbf{x}) > 0$ (strogo dopustna rešitev).

Karush-Kuhn-Tuckerjev izrek

V sedlu $(\mathbf{x}^s, \mathbf{u}^s)$ doseže L svoj minimum po $\mathbf{x} \in \mathbb{R}^d$. Torej morajo biti tu njeni odvodi po $\mathbf{x}$ enaki 0:

$$\frac{\partial L}{\partial x_j}(\mathbf{x}^s, \mathbf{u}^s) = 0 \quad \text{oz.} \quad \nabla_{\mathbf{x}} L(\mathbf{x}^s, \mathbf{u}^s) = \mathbf{0}.$$

Obenem pa v sedlu gotovo velja:

$$\hat{f}(\mathbf{x}^s) = L(\mathbf{x}^s, \mathbf{u}^s) = f(\mathbf{x}^s) + \sum_{i=1}^{l} u_i^s f_i(\mathbf{x}^s);$$

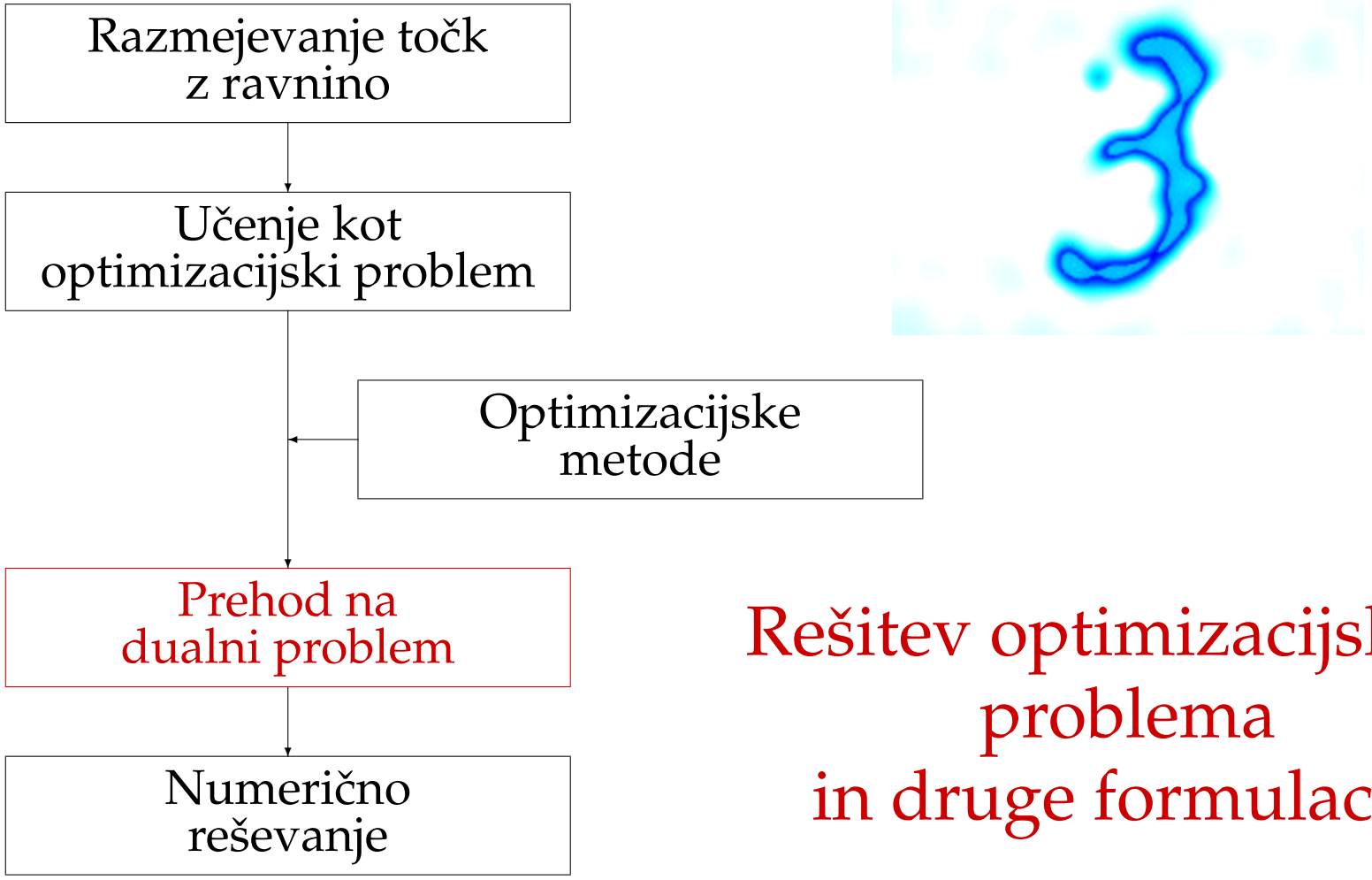
in $\hat{f}(\mathbf{x}^s) = f(\mathbf{x}^s)$, ker je $\mathbf{x}^s$ dopustna rešitev;
torej je $\sum_{i=1}^{l} u_i^s f_i(\mathbf{x}^s) = 0$.

Toda ker so Lagrangeovi multiplikatorji nenegativni, je $u_i^s \geq 0$;
in ker je $\mathbf{x}^s$ dopustna rešitev, je $f_i(\mathbf{x}^s) \geq 0$;
zato je zgornji pogoj izpolnjen le, če je

$$\forall i \in 1..l : u_i^s \, f_i(\mathbf{x}^s) = 0.$$

Z besedami: *ali v pogoju velja enakost ali pa je ustrezni Lagrangeov multiplikator enak* 0.

Če naloga ustreza Slaterjevim pogojem, velja izrek tudi v obratno smer: če za nek $(\mathbf{x}^s, \mathbf{u}^s)$ velja $\nabla_{\mathbf{x}} L(\mathbf{x}^s, \mathbf{u}^s) = \mathbf{0}$ in $u_i^s f_i(\mathbf{x}^s) = 0$ in $u_i^s \geq 0$, je tu sedlo in optimum.



Rešitev optimizacijskega problema in druge formulacije

Optimizacija

Naš optimizacijski problem:

$$\begin{aligned} &\text{minimiziraj} \;\; f(\mathbf{w}, b, \boldsymbol{\xi}) := \tfrac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l}\xi_i \\ &\text{pri pogojih} \;\; \forall i \in 1..l : y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned}$$

Vpeljimo Lagrangeove multiplikatorje $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_l)$ in $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_l)$.

$$L(\underbrace{\mathbf{w}, b, \boldsymbol{\xi}}_{„\mathbf{x}“}, \underbrace{\boldsymbol{\alpha}, \boldsymbol{\mu}}_{„\mathbf{u}“}) = \tfrac{1}{2}\sum_{j=1}^{d} w_j^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{l}\mu_i\xi_i.$$

Iščemo pa $g(\boldsymbol{\alpha}, \boldsymbol{\mu}) = \min_{\mathbf{w},b,\boldsymbol{\xi}} L(\mathbf{w}, b, \boldsymbol{\xi})$, zato poglejmo, kje so odvodi L-ja po w_j-jih, b-ju in $\mathbf{x}_i$-jih enaki 0.

$$\begin{aligned} \partial L/\partial w_j &= w_j - \textstyle\sum_{i=1}^{l}\alpha_i y_i x_{ij} = 0 \quad &\Rightarrow \quad \mathbf{w} &= \textstyle\sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i, \\ \partial L/\partial b &= -\textstyle\sum_{i=1}^{l}\alpha_i y_i = 0 \quad &\Rightarrow \quad \textstyle\sum_{i=1}^{l}\alpha_i y_i &= 0, \\ \partial L/\partial \xi_i &= C - \alpha_i - \mu_i = 0 \quad &\Rightarrow \quad \alpha_i + \mu_i &= C. \end{aligned}$$

Upoštevajmo te pogoje v L.

Prehod k dualni nalogi

Upoštevajmo: $\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i$, $\sum_{i=1}^{l} \alpha_i y_i = 0$, $\alpha_i + \mu_i = C$.

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i[y_i(\mathbf{w}^T\mathbf{x}_i - b) - 1 + \xi_i] - \sum_{i=1}^{l}\mu_i\xi_i =$$

$$\tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_j^T\mathbf{x}_i + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i[y_i(\sum_{j=1}^{l}\alpha_j y_j \mathbf{x}_j^T\mathbf{x}_i - b) - 1 + \xi_i] - \sum_{i=1}^{l}\mu_i\xi_i =$$

$$\tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_j^T\mathbf{x}_i - \sum_{i=1}^{l}\sum_{i=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_j^T\mathbf{x}_i + \sum_{i=1}^{l}\alpha_i y_i b + \sum_{i=1}^{l}\alpha_i + \sum_{i=1}^{l}(C - \alpha_i - \mu_i)\xi_i =$$

$$-\tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_j^T\mathbf{x}_i - b\sum_{i=1}^{l}\alpha_i y_i + \sum_{i=1}^{l}\alpha_i + \sum_{i=1}^{l}(C - \alpha_i - \mu_i)\xi_i =$$

$$-\tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_j^T\mathbf{x}_i + \sum_{i=1}^{l}\alpha_i.$$

Dobili smo *dualno nalogo*:

$$\begin{aligned} &\text{maksimiziraj } -\tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_j^T\mathbf{x}_i + \sum_{i=1}^{l}\alpha_i \\ &\text{pri pogojih } \forall i \in 1..l : 0 \le \alpha_i \le C \quad \text{in} \quad \sum_{i=1}^{l}\alpha_i y_i = 0. \end{aligned}$$

Dualna naloga

Če obrnemo predznak kriterijski funkciji:

$$\begin{aligned} &\text{minimiziraj } g(\boldsymbol{\alpha}) := \tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j - \sum_{i=1}^{l}\alpha_i \\ &\text{pri pogojih } \forall i \in 1..l : 0 \le \alpha_i \le C \quad \text{in} \quad \sum_{i=1}^{l}\alpha_i y_i = 0. \end{aligned}$$

To je problem *kvadratnega programiranja*.

Kriterijsko funkcijo lahko zapišemo tudi v matrični obliki:

$$g(\boldsymbol{\alpha}) = \tfrac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha},$$

kjer je Q matrika z elementi $Q_{ij} = y_i y_j \mathbf{x}_i^T\mathbf{x}_j$,
$\mathbf{1} = (1, 1, \ldots, 1)^T$ pa vektor samih enic.

Ko poiščemo optimalno $\boldsymbol{\alpha}$, lahko izrazimo normalo na ravnino po formuli

$$\mathbf{w} = \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i.$$

Kako pridemo do praga b, bomo videli kasneje.
Lahko bi izrazili tudi $\boldsymbol{\xi}$ (ko bi imeli b), a tega za klasifikacijo niti ne potrebujemo.

Podporni vektorji

KKT izrek nam o optimalni rešitvi zagotavlja, za vsak $i \in 1..l$:

$$\alpha_i[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i] = 0 \qquad \text{in} \qquad \mu_i\xi_i = 0.$$

Spomnimo se, da velja še: $\alpha_i + \mu_i = C$, $\alpha_i \geq 0$, $\mu_i \geq 0$, $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i \geq 0$.
Ločimo tri možnosti:

1. $\alpha_i = 0$. Tedaj je $\mu_i = C \neq 0$ in zato $\xi_i = 0$, zato pa $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$. To so vektorji na pravi strani roba.

2. $0 < \alpha_i < C$. Tedaj je $\mu_i = C - \alpha_i > 0$ in zato spet $\xi_i = 0$, vendar pa zdaj $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$. Ti vektorji so na pravi strani roba, vendar ravno še na meji.

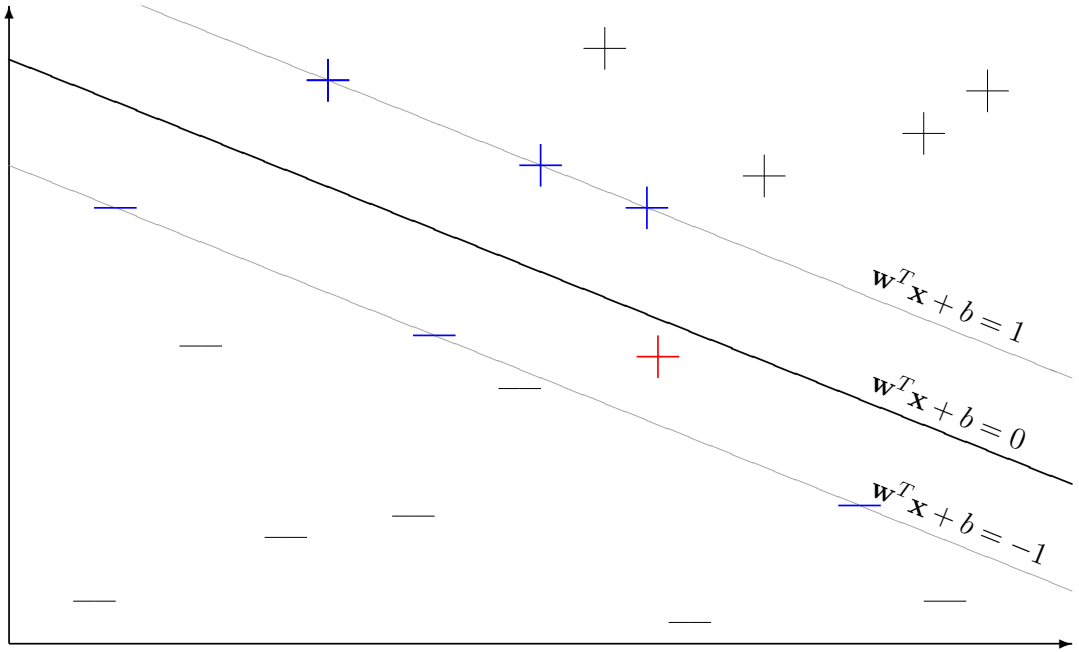
3. $\alpha_i = C$. Tedaj je $\mu_i = 0$ in $y_i(\mathbf{x}^T\mathbf{x}_i + b) \geq 1 - \xi_i$. Zdaj je mogoče, da je $\xi_i > 0$; če je $\xi_i \geq 1$, je tak vektor narobe klasificiran.

V izražavi $\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i$ nastopajo le vektorji iz druge in tretje skupine, ki jim zato pravimo *podporni vektorji*.

- Izkaže se: če bi iz učne množice zavrgli vse ostale vektorje in se ponovno učili, bi dobili isto rešitev.

- Ti vektorji so nekako najbližje razmejitveni ravnini, z obeh strani jo podpirajo, da se ne bi kam premaknila.

Določanje praga b



1. Načeloma za poljuben vektor iz 2. skupine velja $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$, torej $b = y_i - \mathbf{w}^T\mathbf{x}_i$; bolje je vzeti povprečje po vseh teh vektorjih.
2. Ali pa: $b = -\frac{1}{2}\left[\min_{i:\alpha_i<C, y_i=+1} \mathbf{w}^T\mathbf{x}_i + \max_{i:\alpha_i<C, y_i=-1} \mathbf{w}^T\mathbf{x}_i\right]$.
3. Lahko si pomagamo s fitanjem sigmoide.
4. Z validacijsko množico — [CRS02] so opazili, da je lahko to veliko bolje.

Trdi rob (hard margin)

Pri tej različici ne dovolimo napak na učni množici.

Prvotna naloga:

$$\begin{aligned} &\text{minimiziraj} \;\; f(\mathbf{w}, b) = \tfrac{1}{2}||\mathbf{w}||^2 \\ &\text{pri pogojih} \;\; \forall i \in 1..l : y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \end{aligned}$$

Dualna naloga:

$$\begin{aligned} &\text{minimiziraj} \;\; g(\boldsymbol{\alpha}) = \tfrac{1}{2}\alpha^T Q\alpha - \mathbf{1}^T\boldsymbol{\alpha} \\ &\text{pri pogojih} \;\; \forall i \in 1..l : \alpha_i \geq 0 \quad \text{in} \quad \textstyle\sum_{i=1}^{l} \alpha_i y_i = 0. \end{aligned}$$

Tu se lahko zgodi, da prvotna naloga sploh nima dopustnih rešitev. Pri dualni nalogi, kjer α_i zdaj navzgor niso omejene (kot prej s C), gre lahko v takem primeru g prek vseh meja.

Ničeln prag

Pri tej različici fiksiramo prag na $b = 0$, torej se omejimo na ravnine, ki gredo skozi koordinatno izhodišče.

$$\begin{aligned} &\text{minimiziraj } \; f(\mathbf{w}, \boldsymbol{\xi}) = \tfrac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{l} \xi_i \\ &\text{pri pogojih } \; \forall i \in 1..l : y_i \mathbf{w}^T \mathbf{x}_i \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned}$$

S tem odpade iz dualne naloge pogoj $\boldsymbol{\alpha}^T \mathbf{y} = 0$.

$$\begin{aligned} &\text{minimiziraj } \; g(\boldsymbol{\alpha}) = \tfrac{1}{2}\alpha^T Q \alpha - \mathbf{1}^T \boldsymbol{\alpha} \\ &\text{pri pogojih } \; \forall i \in 1..l : 0 \leq \alpha_i \leq C. \end{aligned}$$

V zameno učnim vektorjem pogosto dodamo še eno komponento, ki je pri vseh enaka 1: $\hat{\mathbf{x}}_i = (\mathbf{x}_i, 1)$. Zato ima tudi $\mathbf{w}$ še eno komponento, ki torej učinkuje podobno kot prag: $\hat{\mathbf{w}} = (\mathbf{w}, b)$. Učinek je torej tak, kot če bi imeli:

$$\begin{aligned} &\text{minimiziraj } \; f(\mathbf{w}, b, \boldsymbol{\xi}) = \tfrac{1}{2}(||\mathbf{w}||^2 + b^2) + C \sum_{i=1}^{l} \xi_i \\ &\text{pri pogojih } \; \forall i \in 1..l : y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad \xi_i \geq 0 \end{aligned}$$

Kvadratna kazen

Tu vzamemo vsoto kvadratov kazenskih spremenljivk:

$$\begin{array}{r} \text{minimiziraj} \;\; f(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l} \xi_i^2 \\ \text{pri pogojih} \;\; \forall i \in 1..l : y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \end{array}$$

Omejitve $\xi_i \geq 0$ ne potrebujemo — tako ali tako se ne bi splačalo vzeti negativnega ξ_i, kajti namesto njega je $\xi_i = 0$ tudi dober, pa še f bo manjši.

V dualni nalogi se spremeni matrika drugih odvodov. Po diagonali dobi še vrednosti $1/2C$, kar je numerikom v veliko veselje, ker je tako bolje pogojena:

$$\begin{array}{r} \text{minimiziraj} \;\; g(\boldsymbol{\alpha}) = \frac{1}{2}\alpha^T(Q + \frac{1}{2C}I)\alpha - \mathbf{1}^T\boldsymbol{\alpha} \\ \text{pri pogojih} \;\; \forall i \in 1..l : \alpha_i \geq 0 \quad \text{in} \quad \sum_{i=1}^{l} \alpha_i y_i = 0. \end{array}$$

Tu ni več pogoja $\alpha_i \leq C$ ali česa podobnega, vendar se vseeno ni bati, da bi kakšna α_i ušla v neskončnost, saj imamo v kriterijski funkciji zdaj člen $\frac{1}{4C}\alpha_i^2$.

To radi kombinirajo z ničelnim pragom, da se znebijo nadležnega pogoja z enakostjo.

Izbira parametra C

$$\begin{aligned} &\text{minimiziraj } f(\mathbf{w}, b, \boldsymbol{\xi}) := \tfrac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l}\xi_i \\ &\text{pri pogojih } \forall i \in 1..l : y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned}$$

- C mora posredovati med
  - težnjami k širšemu robu (spomnimo se, da je rob širok $2/||\mathbf{w}||$) in
  - težnjami po čim manjših napakah (ξ_i) na učni množici.
- Ker je v pogojih 1 fiksna, se tudi ξ_i gibljejo tam nekje: $\xi_i = 2$ vedno pomeni napako za celotno širino roba, ipd. Za $||\mathbf{w}||^2$ pa ni tovrstnih omejitev.

  Če se vsi $\mathbf{x}_i$ pomnožijo z λ, mi pa bi želeli „isto ravnino“ kot prej, bo $\hat{\mathbf{w}} = (1/\lambda)\mathbf{w}$ ustrezala pogojem z istimi ξ_i in b.
  - Toda v kriterijski funkciji je zdaj prvi člen λ^2-krat manjši, torej bi čutili napake veliko huje.
  - Zato bi morali vzeti $\hat{C} = C/\lambda^2$.

  Do istega problema pride pri jedrih, kjer učne vektorje tudi nekako preslikamo.

Izbira parametra C

Teorija:

- Recimo, da *vsi naši vektorji*, učni in kasneje testni, *ležijo v krogli s* središčem v koordinatnem izhodišču in *polmerom* R. Za neko konstanto k velja:
- Naj bo f poljuben klasifikator oblike $napoved(\mathbf{x}) = \operatorname{sgn} \mathbf{w}^T\mathbf{x}$;
- naj bodo ξ_i njegove napake na učni množici, tako da je $y_i(\mathbf{w}^T\mathbf{x}) \geq 1 - \xi_i$;
- potem z verjetnostjo vsaj $1 - \delta$ velja:

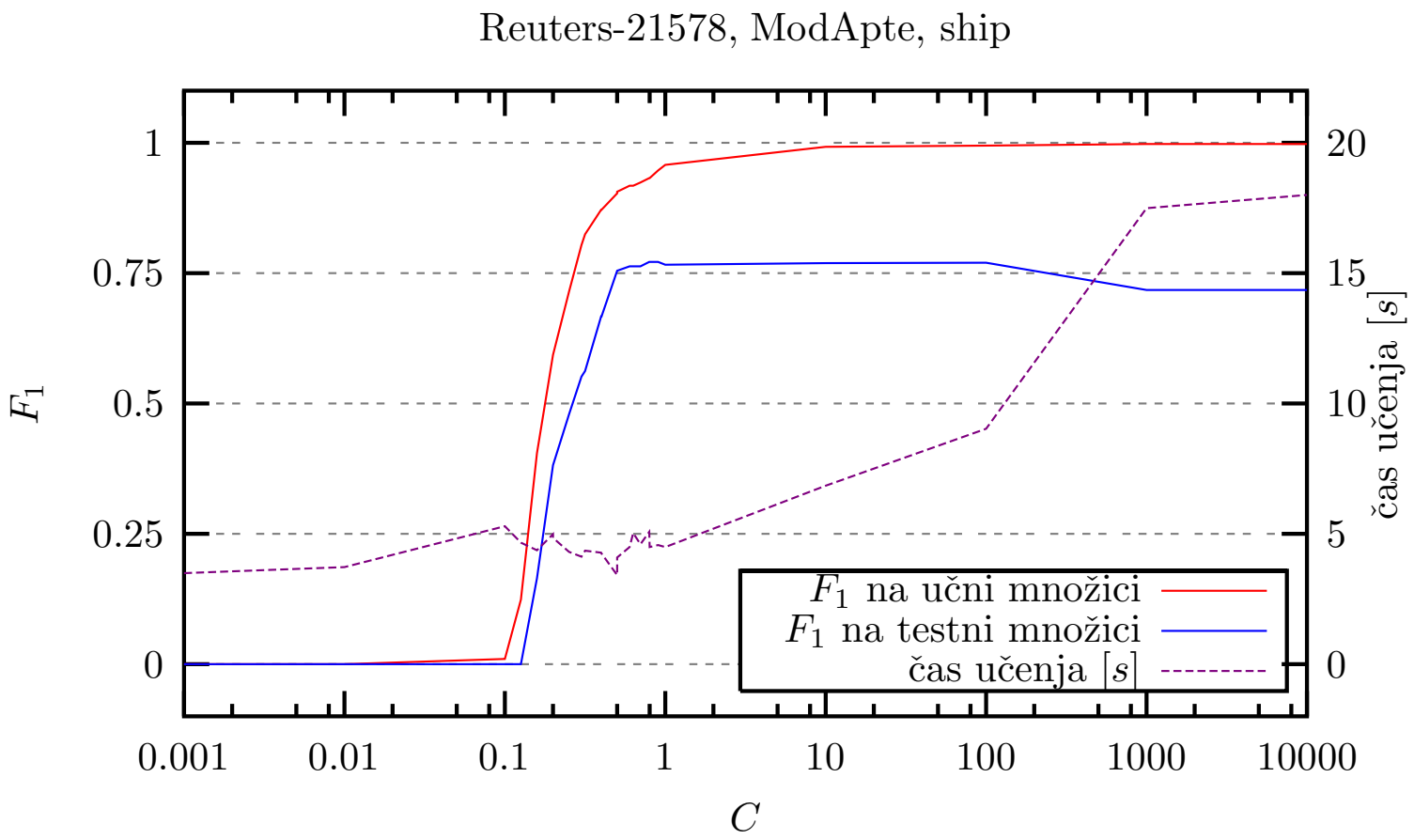
$$verjetnost\ napake(f) \leq \frac{k}{l}\left((||\mathbf{w}||^2R^2 + ||\boldsymbol{\xi}||^2)\log^2 l - \log\delta\right).$$

- Edino, na kar lahko vplivamo, je $||\mathbf{w}||^2R^2 + ||\boldsymbol{\xi}||^2$, kar je isto kot $R^2(||\mathbf{w}||^2 + R^{-2}||\boldsymbol{\xi}||^2)$. Mi pri učenju minimiziramo $||\mathbf{w}||^2 + C||\boldsymbol{\xi}||^2$; desna stran bo torej najmanjša pri $C = R^{-2}$.

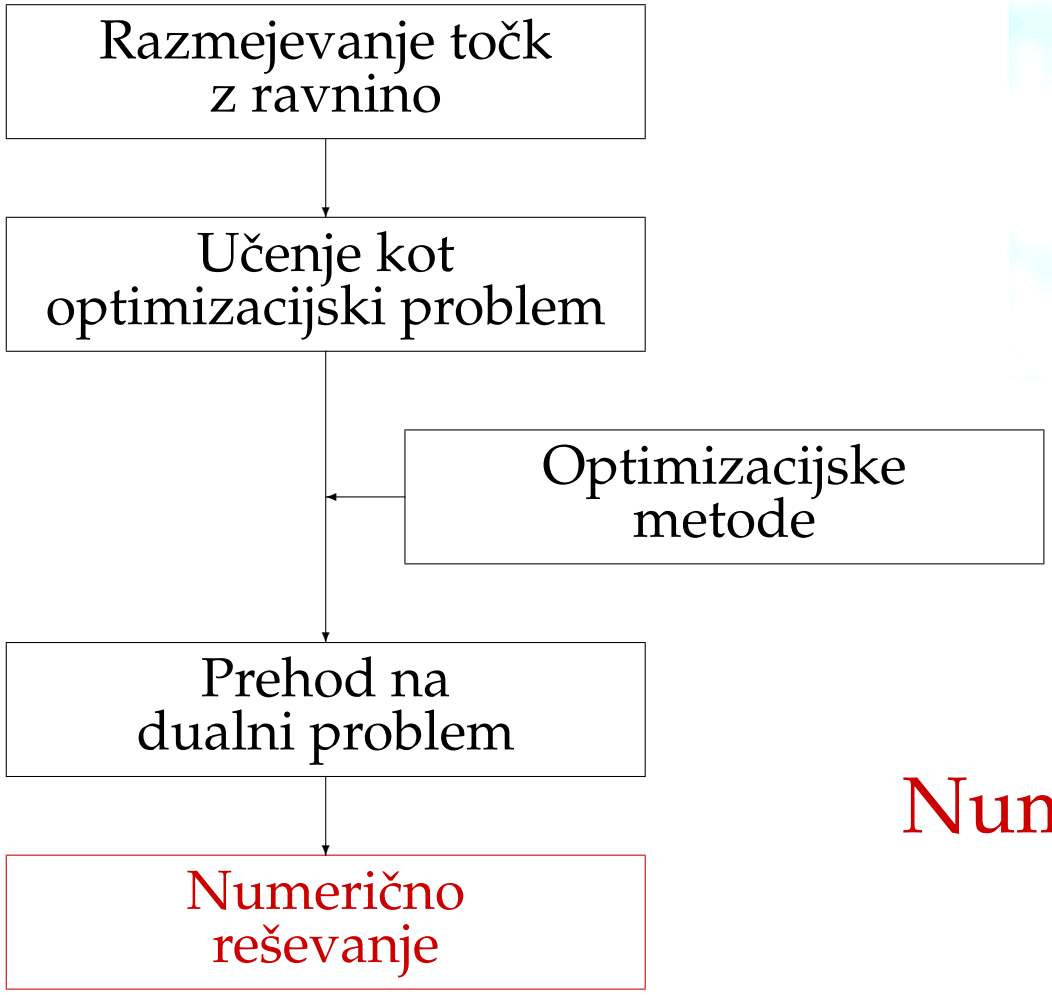
  Ta nasvet sicer predpostavlja ničeln prag b in kvadratno kazen pri optimizaciji, vendar je tudi pri linearni lahko koristen.

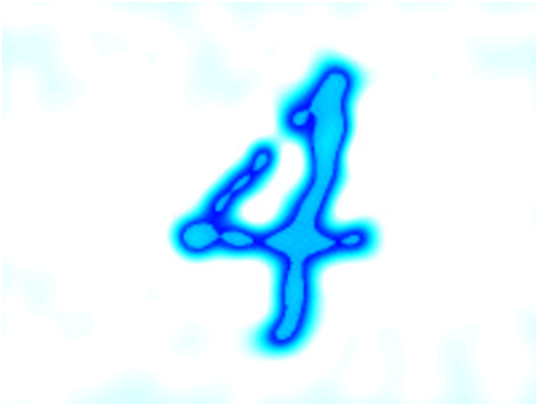
V praksi: *prečno preverjanje*.

Izbira parametra C



- Zelo neugodno je, če vzamemo premajhen C.
- Če vzamemo prevelik C, nam to ne škoduje tako zelo, pač pa učenje po nepotrebnem traja dlje.





Numerično reševanje

Kvadratno programiranje

Numerično moramo rešiti nalogo

$$\begin{aligned} &\text{minimiziraj } g(\boldsymbol{\alpha}) = \tfrac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha} \\ &\text{pri pogojih } \forall i \in 1..l : \alpha_i \geq 0, \quad \textstyle\sum_i y_i \alpha_i = 0. \end{aligned}$$

- Matematiki že dolgo poznajo razne postopke za to, vendar večinoma odpovejo pri velikih problemih.
- Pri nas je l število učnih primerkov in Q je matrika reda $l \times l$, tako da pogosto niti ne gre cela hkrati v pomnilnik.
- Vendar pa, če nekaj spremenljivk fiksiramo, nam ostane problem enake oblike na manj spremenljivkah. Zato lahko standardne postopke uporabimo tako, da fiksiramo večino spremenljivk in optimiziramo le po preostalih (*delovna množica*). Potem optimiziramo po neki drugi skupini spremenljivk in tako naprej.

S KKT pogoji lahko kadarkoli preverimo, če smo že pri optimalni rešitvi. Za vsak i mora veljati:

$$\begin{aligned} \alpha_i = 0 &\implies y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \\ 0 < \alpha_i < C &\implies y_i(\mathbf{w}^T \mathbf{x}_i + b) = 1 \\ \alpha_i = C &\implies y_i(\mathbf{w}^T \mathbf{x}_i + b) \leq 1 \end{aligned}$$

Chunking

To je predlagal že Vapnik (1979):

1. Izberimo prvih nekaj spremenljivk kot delovno množico in optimizirajmo po njih.
2. Ko to naredimo, odstranimo iz delovne množice tiste z $\alpha_i = 0$ in dodajmo vanjo tiste, ki prej niso bile v DM, se pa trenutna ravnina na njih zmoti (imajo $y_i(\mathbf{w}^T\mathbf{x}_i + b) < 1$).
3. Spet optimiziramo na novi DM in to ponavljamo.

Tu se zanašamo na dejstvo, da se rezultat ne spremeni, če iz učne množice zavržemo vse ne-podporne vektorje. Torej je dovolj, če se učimo samo na podpornih; z gornjim postopkom skušamo ugotoviti, kateri so podporni.

Različica: *decomposition* (Osuna *et al.*, 1997) — v delovno množico vedno sprejmemo le toliko novih spremenljivk, kolikor smo jih prej odvzeli iz nje. Očitno moramo dovoliti vsaj tako veliko DM, da bodo šli vanjo vsi podporni vektorji.

Joachimsova različica dekompozicije

- Za delovno množico si izberemo q spremenljivk; optimiziramo; to ponavljamo, dokler ne konvergira. Predlaga $q = 10$.

Spremenljivke si izberemo takole: kriterijska funkcija

$$g(\boldsymbol{\alpha}) = \tfrac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha}$$

ima odvode

$$\nabla_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha}) = Q\boldsymbol{\alpha} - \mathbf{1}^T.$$

- Spremenljivke α_i z negativnim odvodom $\partial g / \partial \alpha_i$ bi bilo treba povečevati, tiste s pozitivnim zmanjševati.
- Da bomo lahko spoštovali pogoj $\sum_{i=1}^{l} y_i \alpha_i = 0$, si moramo izbrati nekaj (npr. $q/2$) takih α_i, pri katerih se bo vrednost $y_i \alpha_i$ povečevala, in nekaj takih, pri katerih se bo zmanjševala.
- Pri tem seveda ignoriramo tiste, ki jih ne bi mogli premikati v željeni smeri (če so že na robu intervala $[0, C]$).

Shrinking: če neka α_i veliko iteracij ostaja pri vrednosti 0 (ali C), je v bodoče sploh ne gledamo več. Čez čas preverimo, če je njena vrednost še sprejemljiva.

Zaporedna minimalna optimizacija (SMO)

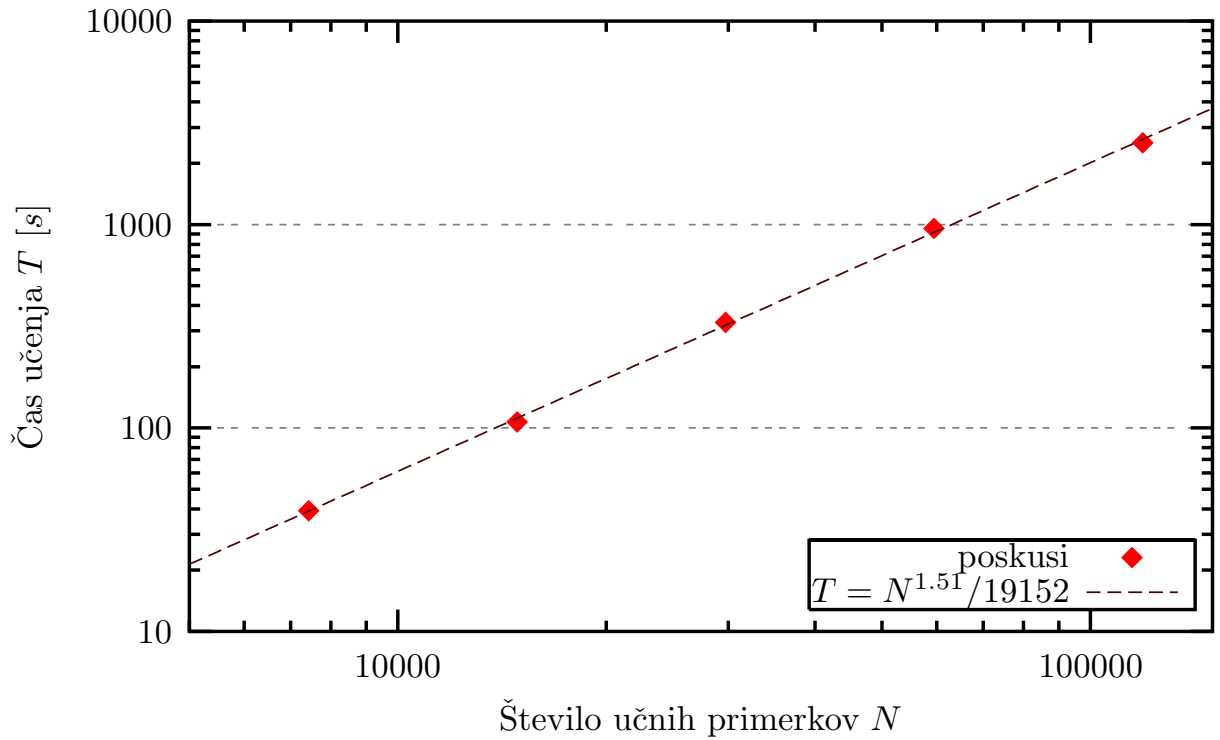
Platt (1999) je opazil:

- Ker imamo pogoj $$\sum_{i=1}^{l} y_i \alpha_i = 0,$$
  ne moremo spreminjati le po ene α_i — moramo vsaj dve naenkrat, da bomo lahko ohranjali to enakost. Izberimo le dve, α_i in α_j.
- Od gornjega pogoja nam ostane zdaj
  $$y_i \alpha_i^{\text{stara}} + y_j \alpha_j^{\text{stara}} = y_i \alpha_i^{\text{nova}} + y_j \alpha_j^{\text{nova}}.$$
  Torej: ko spremenimo npr. α_i v α_i^{nova}, je s tem že točno določena tudi edina sprejemljiva vrednost za α_j.
- Torej imamo v bistvu en sam parameter — spremembo α_i.
  g je kvadratna funkcija tega in brez težav poiščemo minimum.

Hevristika za izbor dveh spremenljivk:

1. α_i naj bo tista, ki najhuje krši svoj KKT pogoj.
2. α_j tista, ki v paru z α_i obeta največji premik.

Časovna zahtevnost učenja SVMjev



- Množica 118924 učnih primerkov in njene podmnožice.
- Linearna regresija pravi:

$$[\textit{čas učenja}] \approx \frac{1}{19152.342}[\textit{število učnih primerkov}]^{1.51725}\ s.$$

- Časovna zahtevnost je med linearno in kvadratno,
  različni avtorji navajajo različne eksponente.

Nelinearni modeli in jedra

Prehod na nelinearne modele

Recimo, da bi pred učenjem vse učne vektorje preslikali v nek nov vektorski prostor $\mathcal{F}$:

$$\begin{aligned} \phi : \ & \mathbb{R}^d \to \mathcal{F} \\ & \mathbf{x} \mapsto \phi(\mathbf{x}) \end{aligned}$$

Če pišemo $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots)$, so $\phi_i(\mathbf{x})$ *oblike* ali *značilke* (*features*) primerka $\mathbf{x}$, $\mathcal{F}$ je prostor značilk (*feature space*).

Naš optimizacijski problem je bil prej

$$\begin{aligned} &\text{minimiziraj} \ \ f(\mathbf{w}, b, \boldsymbol{\xi}) := \tfrac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l} \xi_i \\ &\text{pri pogojih} \ \ \forall i \in 1..l : y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \end{aligned}$$

Zdaj pa je:

$$\begin{aligned} &\text{minimiziraj} \ \ f(\hat{\mathbf{w}}, b, \boldsymbol{\xi}) := \tfrac{1}{2}||\hat{\mathbf{w}}||_{\mathcal{F}}^2 + C\sum_{i=1}^{l} \xi_i \\ &\text{pri pogojih} \ \ \forall i \in 1..l : y_i(\langle \hat{\mathbf{w}}, \phi(\mathbf{x}_i)\rangle_{\mathcal{F}} + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \end{aligned}$$

$\hat{\mathbf{w}}$ živi v $\mathcal{F}$ in $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ je skalarni produkt v $\mathcal{F}$.
$||\hat{\mathbf{w}}||_{\mathcal{F}}^2 = \langle \hat{\mathbf{w}}, \hat{\mathbf{w}} \rangle_{\mathcal{F}}$ je norma v prostoru $\mathcal{F}$.

Prehod na nelinearne modele

Dualna naloga je bila prej:

$$\begin{aligned} &\text{minimiziraj } g(\boldsymbol{\alpha}) := \tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j - \sum_{i=1}^{l}\alpha_i \\ &\text{pri pogojih } \forall i \in 1..l : 0 \leq \alpha_i \leq C \quad \text{in} \quad \sum_{i=1}^{l}\alpha_i y_i = 0. \end{aligned}$$

Zdaj pa je:

$$\begin{aligned} &\text{minimiziraj } g(\boldsymbol{\alpha}) := \tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle_{\mathcal{F}} - \sum_{i=1}^{l}\alpha_i \\ &\text{pri pogojih } \forall i \in 1..l : 0 \leq \alpha_i \leq C \quad \text{in} \quad \sum_{i=1}^{l}\alpha_i y_i = 0. \end{aligned}$$

Prej se je normala izražala kot $\mathbf{w} = \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i$.
Zdaj: $\hat{\mathbf{w}} = \sum_{i=1}^{l}\alpha_i y_i \phi(\mathbf{x}_i)$.

Klasifikator je bil prej:

$$napoved(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}^T\mathbf{x} + b).$$

Zdaj:

$$napoved(\mathbf{x}) = \operatorname{sgn}(\langle\hat{\mathbf{w}},\phi(\mathbf{x})\rangle_{\mathcal{F}} + b) = \operatorname{sgn}(b + \sum_{i=1}^{l}\alpha_i y_i \langle\phi(\mathbf{x}_i),\phi(\mathbf{x})\rangle_{\mathcal{F}}).$$

Jedra

Videli smo: *nikjer nam ni treba eksplicitno delati s slikami* $\phi(\mathbf{x})$.

Dovolj je, če znamo računati skalarne produkte med njimi:

$$K(\mathbf{x}, \hat{\mathbf{x}}) = \langle \phi(\mathbf{x}), \phi(\hat{\mathbf{x}}) \rangle_{\mathcal{F}}.$$

Taki funkciji pravimo *jedro* (*kernel*).

- Lahko bi si najprej izbrali ϕ in $\mathcal{F}$ ter potem poskušali najti primerno K.
  (Takšno, ki jo bo preprosto računati.)
- Pogosto pa kar vzamemo nek K in se moramo le še prepričati, da obstajata neka ϕ in $\mathcal{F}$, ki jima K ustreza.

Zadostni pogoji, da je K jedro

Mercerjev izrek:

- Omejimo se na neko kompaktno (zaprto in omejeno) podmnožico $\mathcal{C} \subset \mathbb{R}^d$.
- g je *kvadratno integrabilna* na $\mathcal{C}$ ntk.

$$\int_{\mathcal{C}} g^2(\mathbf{x}) d\mathbf{x} < \infty.$$

- K naj bo simetrična in za vsaki kvadratno integrabilni f, g naj velja:

$$\iint_{\mathcal{C}\times\mathcal{C}} f(\mathbf{x}) K(\mathbf{x}, \hat{\mathbf{x}}) g(\hat{\mathbf{x}}) d\mathbf{x} d\hat{\mathbf{x}} \geq 0.$$

  Potem je K jedro.

Še en zadosten pogoj:

- Za vsako končno množico $\{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$ mora biti matrika $K = (k_{ij})$ z elementi $k_{ij} := K(\mathbf{x}_i, \mathbf{x}_j)$ pozitivno semidefinitna.

Bolj človeški pogoji

Če so K_1, K_2, K_3 jedra, so tudi tole jedra:

- $K(\mathbf{x}, \hat{\mathbf{x}}) = K_1(\mathbf{x}, \hat{\mathbf{x}}) + K_2(\mathbf{x}, \hat{\mathbf{x}})$
- $K(\mathbf{x}, \hat{\mathbf{x}}) = \lambda K_1(\mathbf{x}, \hat{\mathbf{x}})$ za poljubno $\lambda \in \mathbb{R}$
- $K(\mathbf{x}, \hat{\mathbf{x}}) = K_1(\mathbf{x}, \hat{\mathbf{x}}) K_2(\mathbf{x}, \hat{\mathbf{x}})$
- $K(\mathbf{x}, \hat{\mathbf{x}}) = f(\mathbf{x}) f(\hat{\mathbf{x}})$ za poljubno $f : \mathbb{R}^d \to \mathbb{R}$
- $K(\mathbf{x}, \hat{\mathbf{x}}) = K_3(\phi(\mathbf{x}), \phi(\hat{\mathbf{x}}))$ za poljubno $\phi : \mathbb{R}^d \to \mathbb{R}^m$
- $K(\mathbf{x}, \hat{\mathbf{x}}) = \mathbf{x}^T A \hat{\mathbf{x}}$ za poljubno pozitivno semidefinitno A

Polinomska jedra

$$K(\mathbf{x}, \hat{\mathbf{x}}) = (\mathbf{x}^T \hat{\mathbf{x}} + 1)^p$$

Na primer: recimo, da naši učni vektorji živijo v 2-d prostoru:
$\mathbf{x} = (x_1, x_2)$, $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2)$. Potem je za $p = 2$:

$$\begin{aligned} K(\mathbf{x}, \hat{\mathbf{x}}) &= (x_1\hat{x}_1 + x_2\hat{x}_2 + 1)^2 = \\ &= x_1^2\hat{x}_1^2 + x_2^2\hat{x}_2^2 + 2x_1x_2\hat{x}_1\hat{x}_2 + 2x_1\hat{x}_1 + 2x_2\hat{x}_2 + 1 = \\ &= (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)(\hat{x}_1^2, \hat{x}_2^2, \sqrt{2}\hat{x}_1\hat{x}_2, \sqrt{2}\hat{x}_1, \sqrt{2}\hat{x}_2, 1)^T. \end{aligned}$$

Torej deluje $K(\mathbf{x}, \hat{\mathbf{x}})$ tako, kot da bi oba vektorja preslikali s preslikavo

$$\phi : \mathbb{R}^2 \to \mathbb{R}^6, \qquad \phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)^T$$

in potem v $\mathbb{R}^6$ naredili navaden skalarni produkt.

Ko se naučimo neko normalo $\hat{\mathbf{w}} = (A, B, C, D, E, F)^T \in \mathbb{R}^6$, dobimo klasifikator

$$\textit{napoved}(\mathbf{x}) = \operatorname{sgn}(\hat{\mathbf{w}}^T \phi(\mathbf{x}) + b) = \operatorname{sgn}(Ax_1^2 + Bx_2^2 + Cx_1x_2 + Dx_1 + Ex_2 + F + b).$$

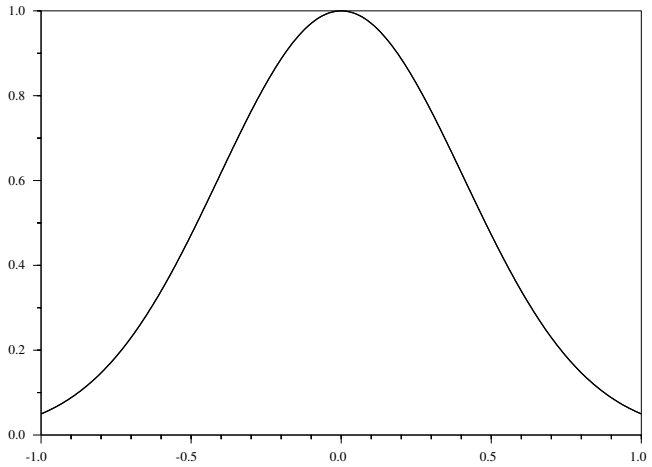
Torej smo 2-d ravnino, v kateri ležijo naši $\mathbf{x}$, razmejili z neko stožernico.

V splošnem je tako, kot da bi ϕ slikala v nek $\binom{p+d}{p}$-razsežni prostor.

Radialna jedra (radialne bazne funkcije)

$$K(\mathbf{x}, \hat{\mathbf{x}}) = \exp\left[-\frac{||\mathbf{x} - \hat{\mathbf{x}}||^2}{2\sigma^2}\right] \text{ ali } \exp\left[-\gamma||\mathbf{x} - \hat{\mathbf{x}}||^2\right]$$

To ustreza preslikavi ϕ v nek ∞-razsežen prostor $\mathcal{F}$.



$$f(t) = \exp(-3t^2)$$

Klasifikator:

$$napoved(\mathbf{x}) = \text{sgn}\left(b + \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x})\right)$$

si lahko predstavljamo takole:

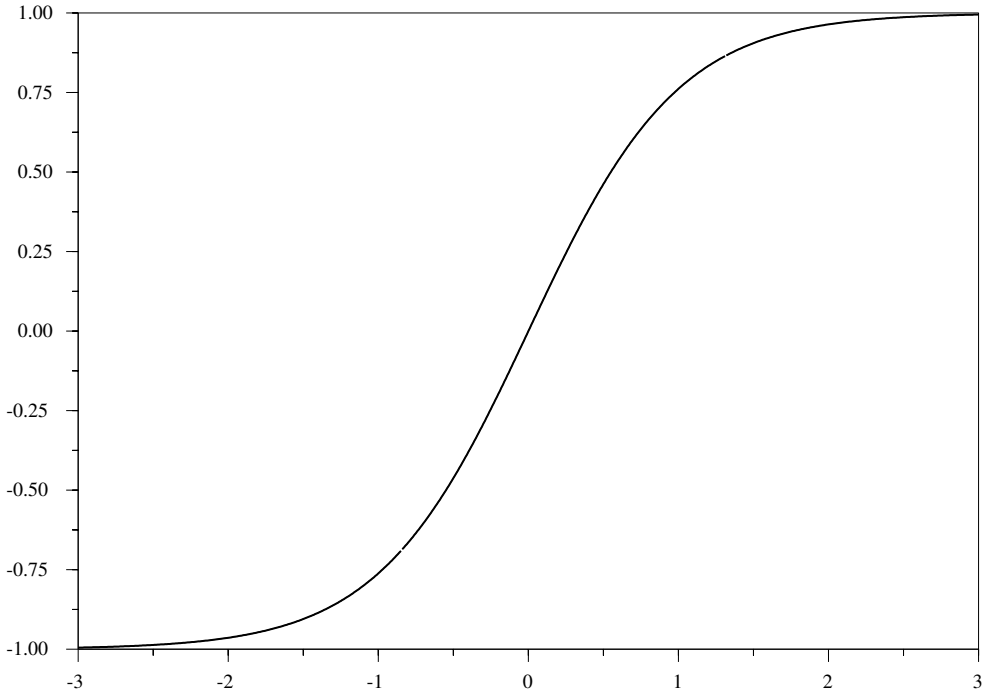
- Vsak učni primerek $\mathbf{x}_i$ glasuje za svoj razred y_i.
  Na koncu vzamemo uteženo vsoto teh glasov in jo primerjamo s pragom $-b$.
- Pri glasovanju damo večjo težo tistim učnim primerkom $\mathbf{x}_i$,
  ki so bliže novemu vektorju $\mathbf{x}$.
- Utež primerka $\mathbf{x}_i$ je $K(\mathbf{x}_i, \mathbf{x})$, torej Gaussova (zvonasta) funkcija razdalje $||\mathbf{x}_i - \mathbf{x}||$. Večja ko je γ, ožji so zvonovi in manjši je vpliv posameznega primerka.

Sigmoidna jedra

$$K(\mathbf{x}, \hat{\mathbf{x}}) = \mathrm{th}(\kappa \mathbf{x}^T \hat{\mathbf{x}} + \Theta)$$

Sploh niso jedra: na primer, $K(\mathbf{x}, \mathbf{x})$ bi moral biti > 0, pa ni, če je $\kappa > 0$, $\Theta < 0$ in

$$||\mathbf{x}|| < \sqrt{-\Theta/\kappa}.$$



Sigmoidna krivulja:

$$\mathrm{th}\, t = \frac{e^t - e^{-t}}{e^t + e^{-t}} = 1 - \frac{2}{e^{2t} + 1}.$$

- Θ določa, kje bo prehod: $f(t) = \mathrm{th}(\kappa t + \Theta)$ ima ničlo pri $t = -\Theta/\kappa$
- κ določa, kako strm bo prehod: f ima v ničli odvod κ

Jedra za nize

Naj bo $x = a_1 a_2 \ldots a_n$ niz n znakov nad abecedo Σ. Niz $u = b_1 b_2 \ldots b_m$ se pojavlja kot podniz v x, če

$$\exists i_1, \ldots, i_m : \qquad 1 \leq i_1 < i_2 < \cdots < i_m \leq n, \quad a_{i_1} = b_1, \ldots, a_{i_m} = b_m.$$

Dajmo tej pojavitvi vrednost $\lambda^{(i_m - i_1 + 1) - m}$ (za neko $\lambda \in [0, 1]$). Definirajmo

$$\phi_u(x) = \sum_{\text{po vseh pojavitvah } u \text{ v } x} \text{vrednost te pojavitve.}$$

In:

$$\phi : \Sigma^* \to \mathbb{R}^{\Sigma^m}, \qquad \phi(x) := \langle \phi_u(x) \rangle_{u \in \Sigma^m}.$$

Potem lahko definiramo jedro:

$$K(x, \hat{x}) = \sum_{u \in \Sigma^m} \phi_u(x) \phi_u(\hat{x})$$

Računamo ga lahko z dinamičnim programiranjem v $O(m \cdot |x| \cdot |\hat{x}|)$.
Menda se je na besedilih obneslo tako dobro kot tradicionalna jedra, včasih še malo bolje (Lodhi *et al.*, 2000).

Posplošena jedra

Pri jedrih dobimo klasifikator oblike $napoved(\mathbf{x}) := \operatorname{sgn}(b + \sum_{j=1}^{l} \alpha_j y_j K(\mathbf{x}, \mathbf{x}_j))$
in pri učenju minimiziramo

$$\begin{aligned} & \tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) + C \sum_{i=1}^{l} \xi_i \\ \text{pri } & \forall i \in 1..l : y_i(b + \textstyle\sum_{j=1}^{l} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)) \geq 1 - \xi_i, \qquad \xi_i \geq 0. \end{aligned}$$

Od tod izvira pogoj, da K ne more biti kakršna koli funkcija — biti mora jedro, torej pozitivno definitna, sicer so lahko pri optimizaciji težave.

Mangasarian je predlagal drugačen optimizacijski problem (*generalized SVM*):

$$\begin{aligned} \text{minimiziraj } & \tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i H_{ij} \alpha_j + C \sum_i \xi_i \\ \text{pri pogojih } & \forall i \in 1..l : y_i(b + \textstyle\sum_{j=1}^{l} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)) \geq 1 - \xi_i, \qquad \xi_i \geq 0. \end{aligned}$$

Ta problem je dovolj lep, čim je $H = (H_{ij})$ pozitivno definitna — ne glede na to, kakšna funkcija je K. Klasifikator je tak kot zgoraj.

Posplošena jedra

Najpreprostejša različica: $H = I$, minimizirajmo $\sum_i \alpha_i^2$.
Kot bi želeli čim preprostejši model (veliko $\alpha_i = 0$).

Optimizacijski problem

$$\begin{aligned} &\text{minimiziraj } \tfrac{1}{2}\textstyle\sum_{i=1}^{l} \alpha_i^2 + C\sum_i \xi_i \\ &\text{pri pogojih } \forall i \in 1..l : y_i(b + \textstyle\sum_{j=1}^{l} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)) \geq 1 - \xi_i, \qquad \xi_i \geq 0. \end{aligned}$$

je ekvivalenten temu, da bi $\mathbf{x}$ preslikali v

$$\phi(\mathbf{x}) := (K(\mathbf{x}, \mathbf{x}_1), K(\mathbf{x}, \mathbf{x}_2), \ldots, K(\mathbf{x}, \mathbf{x}_l)) \in \mathbb{R}^l$$

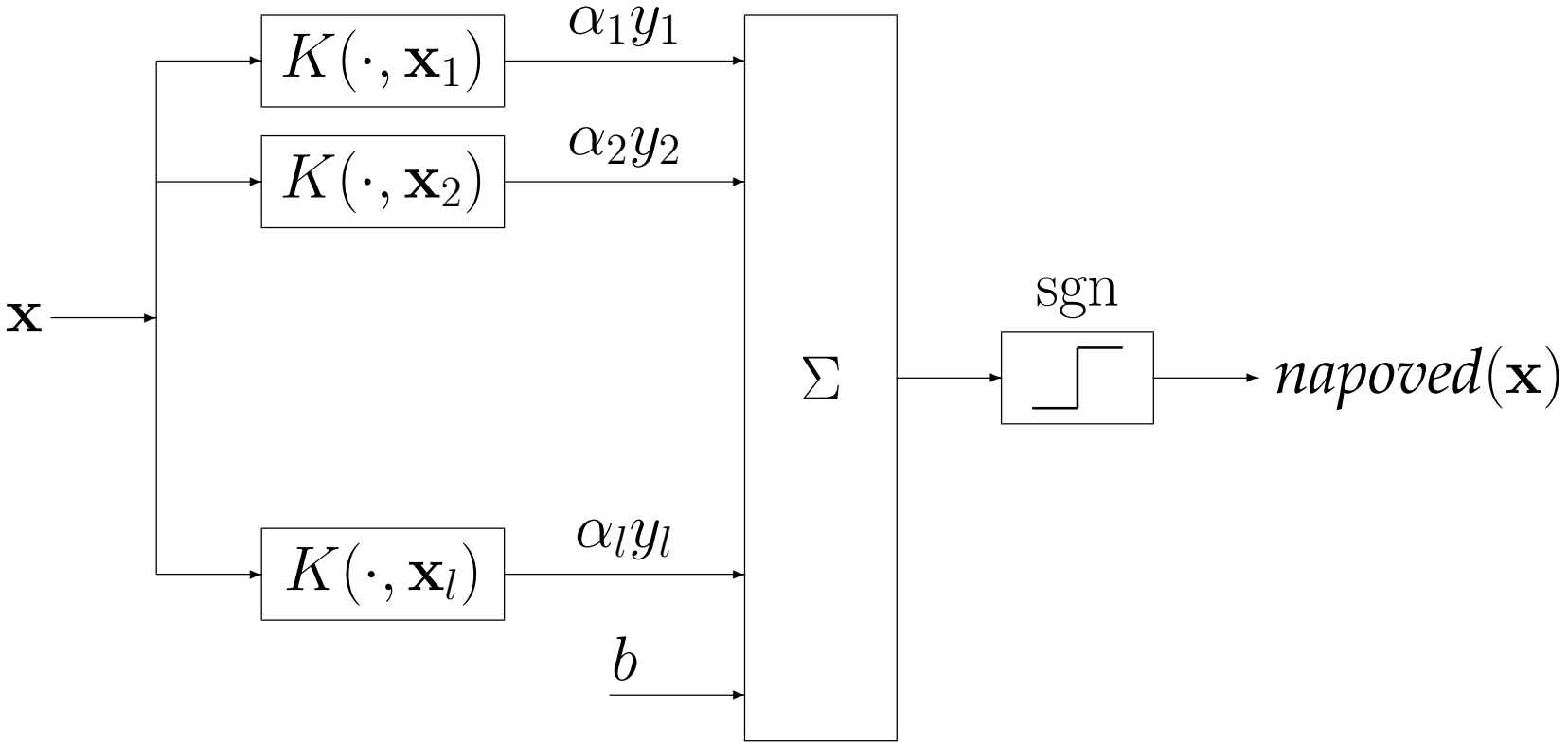
in iskali v prostoru $\mathbb{R}^l$ ravnino oblike $\boldsymbol{\alpha}^T \phi(\mathbf{x}) + b$:

$$\begin{aligned} &\text{minimiziraj } \tfrac{1}{2}||\boldsymbol{\alpha}||^2 + C\textstyle\sum_i \xi_i \\ &\text{pri pogojih } \forall i \in 1..l : y_i(b + \boldsymbol{\alpha}^T \phi(\mathbf{x}_i)) \geq 1 - \xi_i, \qquad \xi_i \geq 0. \end{aligned}$$

Torej je ta formulacija takšna, kot da bi si rekli: primerka $\mathbf{x}$ in $\hat{\mathbf{x}}$ sta si podobna (imata velik $\langle \phi(\mathbf{x}), \phi(\hat{\mathbf{x}}) \rangle_{\mathcal{F}}$), če so vrednosti $K(\mathbf{x}, \mathbf{x}_i)$ približno take kot $K(\hat{\mathbf{x}}, \mathbf{x}_i)$; torej: če imata enak vzorec podobnosti do učnih primerkov ($\mathbf{x}_i$, $i \in 1..l$).

Razno

SVM kot nevronska mreža



- Na prvem nivoju je l celic; i-ta računa $K(\mathbf{x}, \mathbf{x}_i)$.
- Na drugem nivoju je ena celica, ki izračuna uteženo vsoto $b + \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i)$ in vrne njen predznak.

Transduktivni SVM

Tu imamo poleg običajne učne množice še nekaj neoznačenih vektorjev $\mathbf{x}_j^\star$, $j \in 1..m$ (ki pa prihajajo iz iste verjetnostne porazdelitve).

Vapnik (1998) je priporočil:

- *Pripišimo vsakemu od neoznačenih primerkov oznako* ($+1$ ali -1) in to tako, da bo potem pri učenju na obojih skupaj mogoče doseči čim manjšo vrednost kriterijske funkcije.

$$\begin{aligned} \text{minimiziraj} \quad & f(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^\star, \mathbf{y}^\star) := \tfrac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l}\xi_i + C^\star\sum_{j=1}^{m}\xi_i^\star \\ \text{pri pogojih} \quad & \forall i \in 1..l : y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \\ & \forall j \in 1..m : y_j^\star(\mathbf{w}^T\mathbf{x}_j^\star + b) \geq 1 - \xi_j^\star, \quad \xi_j^\star \geq 0, \quad y_j^\star \in \{1, -1\} \end{aligned}$$

Tega ne moremo kar minimizirati, saj so spremenljivke $y_j^\star$ diskretne.

Transduktivni SVM

$$\begin{aligned}
\text{minimiziraj}\quad & f(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^\star, \mathbf{y}^\star) := \tfrac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l}\xi_i + C^\star\sum_{j=1}^{m}\xi_i^\star \\
\text{pri pogojih}\quad & \forall i \in 1..l : y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \\
& \forall j \in 1..m : y_j^\star(\mathbf{w}^T\mathbf{x}_j^\star + b) \geq 1 - \xi_j^\star, \quad \xi_j^\star \geq 0, \quad y_j^\star \in \{1, -1\}
\end{aligned}$$

Zato je Joachims (1999) predlagal *hevristiko*:

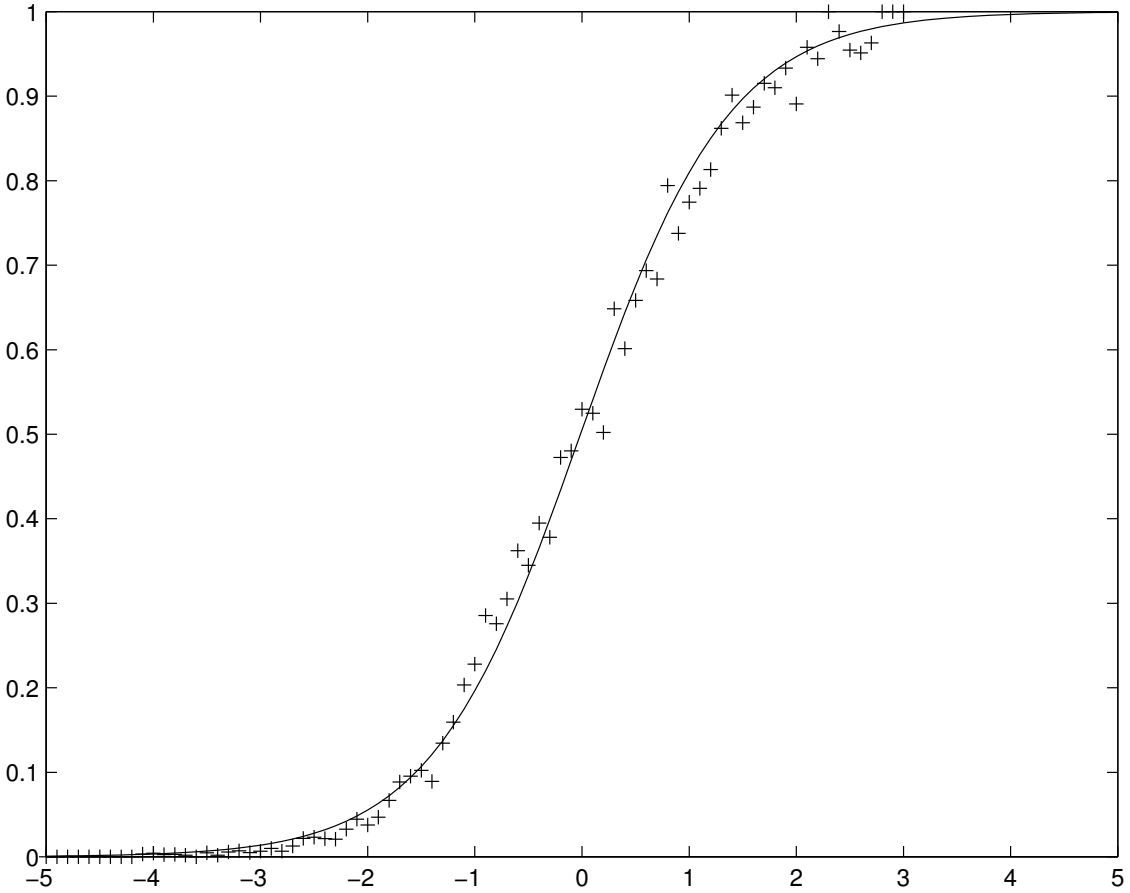
1. Najprej se naučimo le na prvotni učni množici.
   Dobljena $\mathbf{w}$ in b uporabimo, da označimo ostale primerke:
   nekaj $\mathbf{x}_j^\star$ z najvišjimi $\mathbf{w}^T\mathbf{x}_j^\star$ dobi oznako $+1$, ostali -1.

2. Pripravimo nov model na vseh primerkih.
   Dokler gre, poskušamo zamenjati oznaki kakšnih takih $\mathbf{x}_{j_1}^\star$ in $\mathbf{x}_{j_2}^\star$,
   ki sta različno označena in se dosedanji model „zmoti“ na obeh
   (torej: napove nasprotno oznako od tiste, ki smo jima jo pripisali mi).
   Če je ob ponovnem učenju f manjša, zamenjavo obdržimo.

3. Ko to ne gre več, vpliv neoznačenih primerkov ($C^\star$) povečamo in ponovimo
   prejšnjo točko. Ponavljamo, dokler ne postane $C^\star = C$.

Vračanje verjetnosti

Lepo je, če klasifikator vrne *aposteriorno verjetnost* $P(Y = y|X = x)$;
SVM pa vrne le vrednost $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ (oz. njen predznak).



Na abscisi so vrednosti f, na ordinati pa delež pozitivnih primerkov znotraj košarice širine $0{,}1$.

Vračanje verjetnosti

Zato je Platt (1999) predlagal, da bi na f-je položili *sigmoido*:

$$P(Y=1|F=f)=\frac{1}{1+\exp(-Af+B)}, \qquad P(Y=-1|F=f)=1-P(Y=1|F=f).$$

A in B izberimo tako, da se maksimizira $\prod_{i=1}^{l} P(Y=y_i|F=\mathbf{w}^T\mathbf{x}_i+b)$

oziroma logaritem tega. To je verjetnost, da, če dobimo ravno take primerke, kot jih imamo mi tule (torej $\mathbf{x}_i$, $i \in 1..l$), imajo ti primerki tudi ravno take oznake, kot jih vidimo mi (torej y_i).

Koristno je vzeti ločeno validacijsko množico, ne iste kot za učenje prediktorja f.

To nam ponudi tudi malo drugačen klasifikator:

$$napoved(\mathbf{x})=\begin{cases}+1, & \text{če } P(Y=+1|F=\mathbf{w}^T\mathbf{x}+b)\geq 1/2\\ -1, & \text{sicer.}\end{cases}$$

Učinek je tak, kot če bi *spremenili prag*:

$$napoved(\mathbf{x})=\operatorname{sgn}(\mathbf{w}^T\mathbf{x}+b-B/A)$$

To včasih izboljša točnost.

Večrazredni problemi

Imamo r razredov, vsak primerek spada v natanko enega od njih.
Moramo ga preoblikovati v več dvorazrednih problemov.

- *Eden proti ostalim*: naučimo r modelov, ki ločujejo po en razred od ostalih. Če jih več razglasi primerek za pozitivnega, vzamemo tistega z največjo $\mathbf{w}^T\mathbf{x} + b$ (ali pa največjo $P(Y = 1|f)$).
- *Eden proti enemu*: za vsak par razredov naučimo model, ki ju razločuje. Nov primerek pokažemo vsem modelom in se odločimo za razred, ki dobi največ glasov.
- DAGSVM: različica prejšnje strategije. Če model za $\{c_i, c_j\}$ reče, da je primerek v c_i, to sicer še ne pomeni, da je res v c_i, pač pa, da skoraj gotovo ni v c_j.
- *Kodne matrike*: naučimo m modelov; vrednost $M_{ij} \in \{-1, 0, 1\}$ pove, kako uporabimo primerke razreda c_i pri učenju j-tega modela.
  - Nov primerek pokažemo vsem, dobimo vrstico napovedi, ki jo primerjamo z vrsticami matrike M.
  - Razne ideje glede matrike M: gosta, redka, maksimalna, ECC.

Allwein *et al.* (2000) ter Hsu in Lin (2001) priporočajo pristop „eden proti enemu“.

Regresijski SVM

Radi bi napovedovali takole: $y = \langle \mathbf{w}, \mathbf{x} \rangle + b.$

To si lahko predstavljamo kot ravnino v $(d+1)$-razsežnem prostoru. Učni primerki $(\mathbf{x}_i, y_i)$ naj vsi ležijo čim bližje te ravnine.

$$|y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle + b| < \varepsilon$$

Če velja to, ležijo vse $(\mathbf{x}_i, y_i)$ znotraj pasu širine $2\varepsilon/\sqrt{1 + ||\mathbf{w}||^2}$. Torej je spet dobro minimizirati $||\mathbf{w}||$.

$$\begin{aligned} \text{minimiziraj} \quad & \tfrac{1}{2}||\mathbf{w}||^2 + C \textstyle\sum_{i=1}^{l} (\xi_i^{\oplus} + \xi_i^{\ominus}) \\ \text{pri pogojih } (\forall i \in 1..l): \quad & y_i \leq \langle \mathbf{w}, \mathbf{x}_i \rangle + b + \varepsilon + \xi_i^{\oplus} \\ & y_i \geq \langle \mathbf{w}, \mathbf{x}_i \rangle + b - \varepsilon - \xi_i^{\ominus} \\ & \xi_i^{\oplus} \geq 0, \quad \xi_i^{\ominus} \geq 0 \end{aligned}$$

ε si izberemo vnaprej — za majhne napake se sploh ne zmenimo.

To spet lahko predelamo v dualni problem in rešujemo s kvadratnim programiranjem.

ν-SVM (Schölkopf *et al.*, 1998)

Namesto parametra C uporabimo $\nu \in [0, 1]$:

$$\begin{aligned} \text{minimiziraj} \quad & f(\mathbf{w}, b, \gamma, \boldsymbol{\xi}) := \tfrac{1}{2}||\mathbf{w}||^2 - \nu\gamma + \tfrac{1}{l}\sum_{i=1}^{l} \xi_i \\ \text{pri pogojih } (\forall i \in 1..l): \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq \gamma - \xi_i \\ & \xi_i \geq 0, \quad \gamma \geq 0 \end{aligned}$$

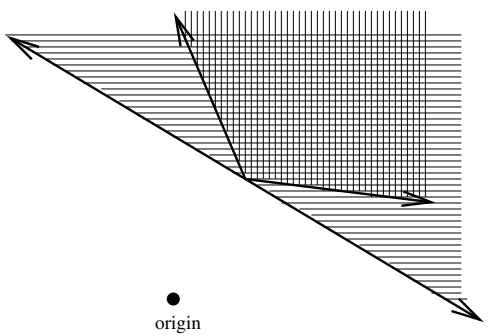
Širina roba je zdaj $2\gamma/||\mathbf{w}||$. Pokazati se da:

- delež učnih primerkov, ki niso na pravi strani roba $\leq \nu$
- delež podpornih vektorjev $\geq \nu$
- če bi dobivali vse več učnih podatkov in če porazdelitev, iz katere prihajajo, ni preveč čudna, bi v gornjih dveh neenačbah konvergirali k enakosti.

S parametrom ν nadziramo delež podpornih vektorjev in napak, tako kot s C v prvotni formulaciji, le da z ν bolj neposredno.

ROMMA (relaxed on-line maximum margin algorithm, Li & Long, 2002)

- On-line učenje: sproti, ko prihajajo novi učni primerki, želimo popravljati model.



- Omejimo se na $b = 0$ in trdi rob. SVMjev optimizacijski problem bi bil:

$$\begin{aligned} &\text{minimiziraj } \tfrac{1}{2}||\mathbf{w}||^2 \\ &\text{pri pogojih } (\forall i \in 1..l): \; y_i \mathbf{w}^T \mathbf{x}_i \geq 1 \end{aligned}$$

  Iščemo torej tisto točko $\mathbf{w}$ iz dopustne množice, ki je najbližja koordinatnemu izhodišču.

- Več ko je primerkov, bolj je dopustna množica zapletena.

- Naj bo $\mathbf{w}_1$ najkrajša normala v dopustni množici.
  Če zamenjamo vse pogoje z enim samim:

$$\mathbf{w}^T \mathbf{w}_1 \geq ||\mathbf{w}_1||^2,$$

  je nova dopustna množica nadmnožica stare,
  najkrajši $\mathbf{w}$ v njej pa je še vedno $\mathbf{w}_1$.

ROMMA

Postopek je torej tak:

- Prvi učni primerek nam da kot pogoj le eno linearno neenačbo: $y_1 \mathbf{w}^T \mathbf{x}_1 \geq 1$
- Za vsak naslednji učni primerek $(\mathbf{x}_t, y_t)$:
  - Dosedanji neenačbi $\mathbf{w}^T \mathbf{u}_{t-1} \geq v_{t-1}$
    se pridruži še tista od trenutnega učnega primerka: $y_t \mathbf{w}^T \mathbf{x}_t \geq 1.$
  - Izračunajmo najkrajši $\mathbf{w}$, ki ustreza obema; recimo mu $\mathbf{w}_t$.
  - Zamenjajmo obe neenačbi z eno samo: $\mathbf{w}^T \mathbf{w}_t \geq ||\mathbf{w}_t||^2.$
    Torej imamo $\mathbf{u}_t = \mathbf{w}_t, v_t = ||\mathbf{w}_t||^2.$

Vpeljati je mogoče tudi neke vrste kvadratno kazen ($+C \sum_i \xi_i^2$).

- Jedru po diagonali prištejemo $1/2C$:

$$K_{\text{novo}}(\mathbf{x}_i, \mathbf{x}_j) := K_{\text{staro}}(\mathbf{x}_i, \mathbf{x}_j) + (1/2C) \cdot \delta_{ij}.$$

- Pri linearnem jedru je to enako, kot če bi vsak $\mathbf{x}_i$ podaljšali z l komponentami, ki so same ničle, razen i-te, ki je enica.
  Očitno je, da je problem potem res linearno separabilen.

Programi

SvmLight (T. Joachims):

- Posebej hiter pri linearnem jedru.
- Podpira običajno in transduktivno formulacijo klasifikacijskega SVM, v zadnji verziji podpira tudi regresijo.

LibSvm (C.-C. Chang, C.-J. Lin):

- Poleg običajnega SVM podpira še ν-SVM, regresijo in ocenjevanje podpore porazdelitev.
- Podpira večrazredne probleme (strategija „eden proti enemu“).

MySvm (S. Rüping), itd., itd.

Demo v javi: *http://svm.cs.rhul.ac.uk/pagesnew/GPat.shtml*

Literatura

- C. J. C. Burges: *A tutorial on support vector machines for pattern recognition*. Data Mining and Knowledge Discovery, 2(2):121–167, June 1998.

Delavnice na NIPS (Neural Information Processing Systems):

- NIPS '97: B. Schölkopf, C. J. C. Burges, A. J. Smola (eds.): *Advances in kernel methods: Support vector learning*. MIT Press, 1998.
- NIPS '98: A. J. Smola, P. J. Bartlett, B. Schölkopf, D. Schuurmans: *Advances in large-margin classifiers*. MIT Press, 2000.
- NIPS '99: Machine Learning, special issue on SVMs and kernel methods, 46(1–3), January–March 2002.
- NIPS 2000: Journal of Machine Learning Research, special issue on kernel methods, December 2001.

Literatura

Knjige:

- N. Christianini, J. Shawe-Taylor: *An introduction to Support Vector Machines and other kernel-based methods*. Cambridge University Press, 1999.

  [*www.support-vector.net*]

- B. Schölkopf, A. J. Smola: *Learning with kernels*. MIT Press, 2001.
  [*www.learning-with-kernels.org*]

- R. Herbrich: *Learning kernel classifiers*. MIT Press, 2001.
  [*www.learning-kernel-classifiers.org*]

- V. N. Vapnik: *Statistical learning theory*. Wiley-Interscience, 1998.

- V. N. Vapnik: *The nature of statistical learning theory*. Springer, 1995, ^{2}1999.

Web site:

- *www.kernel-machines.org*