8. Stroj potpornih vektora

Strojno učenje 1, UNIZG FER, ak. god. 2022./2023.

Jan Šnajder, predavanja, v3.1

Prošla tri tjedna bavili smo se **poopćenim linearnim modelima** za regresiju i klasifikaciju. Danas krećemo u nešto skroz novo i drugačije: govorit ćemo o algoritmu koji se zove **stroj potpornih vektora** (engl. *support vector machine,* **SVM**). Radi se o vrlo učinkovitom klasifikacijskom i regresijskom algoritmu koji je, od devedesetih godina kada je osmišljen, dugo vremena dominirao na sceni strojnog učenja i pobudio velik interes za tzv. **jezgrene metode**. I danas, gotovo trideset godina kasnije, to je i dalje jedan od omiljenih algoritama u teoriji i praksi.

Današnja je tema poprilično sofisticirana i može se ispričati na nekoliko načina. Mi ćemo ići "klasičnim" putem: formalizirat ćemo najprije problem **maksimalne margine**, i onda doći do tzv. **problema kvadratnog programiranja**. U tom trenutku preći ćemo iz tzv. **primarne** formulacije problema u **dualnu** formulaciju. Sve ovo oslanja se na teoriju iz **konveksne optimizacije**, u koju ćemo danas malo dublje uroniti, i, nadam se, izroniti. Sve u svemu, danas će nam biti dosta intenzivno i zanimljivo.

1 Problem maksimalne margine

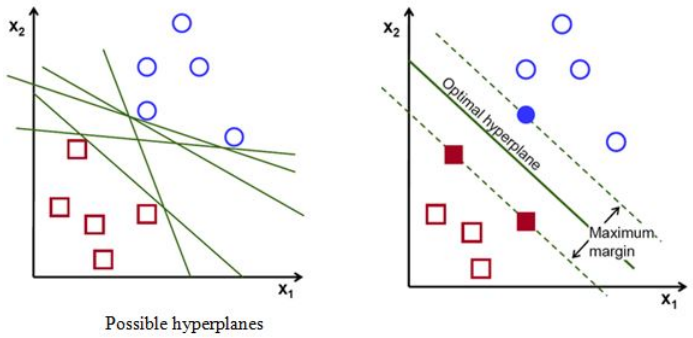
Krenimo od najednostavnije stvari: **modela**. Model SVM-a običan je **linearan model**:

$$h(\mathbf{x};\mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

Primijetite da ovdje nemamo (nelinarnu) aktivacijsku funkciju f. No primijetite da, kao i do sada, možemo upotrijebiti trik s funkcijom preslikavanja $\boldsymbol{\phi}$ kako bismo ostvarili nelinearnost u prostoru primjera.

U nastavku ćemo pretpostaviti da su primjeri **linearno odvojivi**, bilo u ulaznom prostoru ili u prostoru značajki (nakon preslikavanja), pa nećemo pisati funkciju $\boldsymbol{\phi}$, radi jednostavnosti. Ovo je samo naša radna pretpostavka; ona nije realna, i kasnije ćemo je relaksirati. No, za sada, bit će lakše objasniti ideju SVM-a ako pretpostavimo da su primjeri linearno odvojivi.

SVM se temelji na ideji **maksimalne margine**. Objasnimo na primjeru o čemu se radi:

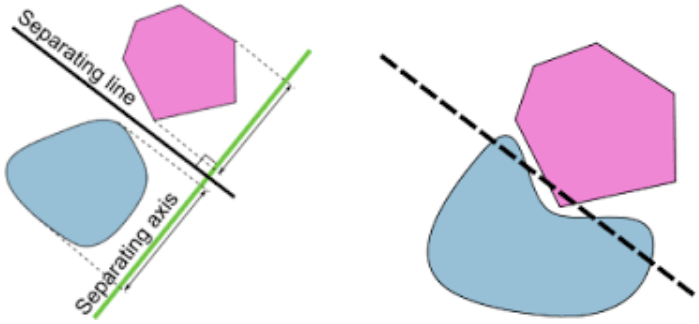


Na lijevoj je slici prikazan dvodimenzijski ulazni prostor sa $N = 10$ primjera iz dvije klase. Kao što smo se dogovorili, primjeri su linearno odvojivi. Zeleni pravci su moguće granice između

klasa, tj. hipoteze koje daju savršenu klasifikaciju na skupu $\mathcal{D}$. Koliko takvih hipoteza postoji? Ako je ulazni prostor $\mathcal{X} = \mathbb{R}^2$, onda takvih hipoteza ima beskonačno mnogo hipoteza. Drugim riječima, beskonačno je mnogo pravaca koje možemo ucrtati između primjera iz ovih dviju klasa. Sada se možemo pitati: u nedostatku bilo kakve druge informacije, koji od tih beskonačno mnogo pravaca bismo trebali preferirati, ako želimo da model dobro generalizira? Kako pozicionirati pravac između primjera iz dvije klase, a da naš model najbolje radi na neviđenim primjerima? Ili, rečeno drugačije: gdje postaviti pravac, a da budemo što manje pristrani? Intuitivno znamo da bi najpametnije bilo da pravac postavimo točno u sredinu, tako da je maksimalno udaljen i od primjera jedne klase i od primjera druge klase, budući da bismo u suprotnom bili pristrani prema jednoj od dviju klasa.

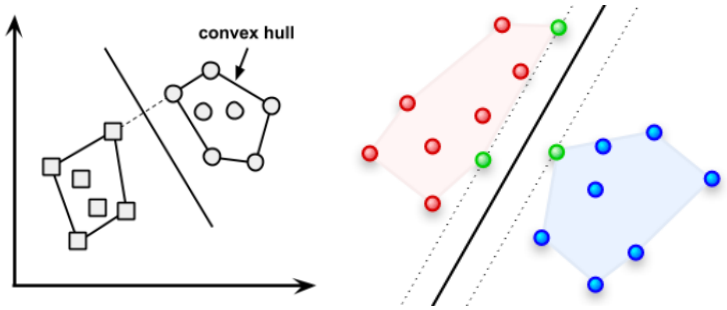
Upravo to je ideja SVM-a: postaviti hiperravninu tako da bude najviše udaljena od primjera iz dviju klasa. Udaljenost od hiperravnine do najbližeg primjera sa svake strane zvat ćemo **margina**. Mi dakle želimo takvu hiperravninu koja **maksimizira marginu**. Intuitivno nam je jasno da će hiperravnina koja maksimizira marginu dobro generalizirati. No to se može pokazati i formalno, u okviru **teorije računalnog učenja** (engl. *computational learning theory, COLT*), što mi, međutim, nećemo raditi. 3

Jasno je da, ako su primjeri linearno odvojivi, onda mora postoji razdvajajuća hiperravnina. No, korisno je to razmotriti sa stajališta geometrije. U geometriji, koncept linearne odvojivosti lijepo odgovara konceptu **konveksnosti skupova**. Konkretno, prema **teoremu o odvajanju hiperravninom** (engl. *hyperplane separation theorem*) za dva **disjunktna i konveksna podskupa** u $\mathbb{R}^n$ postoji hiperravnina koja ih razdvaja. Ako takva hiperravnina ne postoji, onda to znači da barem jedan od ta dva podskupa nije konveksan. 4 5



Lijeva slika prikazuje dva konveksna skupa. Između njih je moguće provući pravac koji ih razdvaja, tj. koji ne prolazi kroz niti jedan od ta dva skupa. Na desnoj slici jedan skup nije konveksan, pa takav razdvajajući pravac ne mora nužno postojati (može, ali ne mora). U našem slučaju, ovi konveksni skupovi odgovaraju tzv. **konveksnim ljuskama** (engl. *convex hull*) primjera iz dviju klasa. Konveksna ljuska je najmanji konveksni skup koji sadrži sve primjere dotične klase. U dvodimenzijskome prostoru to će biti poligon, a općenito n-dimenzijski **politop**. 6

Sad se možemo pitati: koja je veza maksimalne margine i konveksnih ljusaka? Odgovor je da, ako maksimiziramo marginu, onda će hiperravnina zapravo biti **simetrala spojnice** dviju konveksnih ljusaka:



Na lijevoj slici prikazane su dvije konveksne ljuske i spojnica između njih. Spojnica povezuje dvije najbliže točke tih dviju konveksnih ljusaka. Okomica spojnice je pravac koji razdvaja primjere iz dviju klasa. Takav pravac ujedno maksimizira marginu (udaljenost između pravca i konveksnih ljusaka s obje strane), tj. primjer iz prve klase koji je najbliži pravcu i primjer iz druge klase koji je najbliži pravcu su jednako udaljeni od toga pravca. Na desnoj slici prikazana je slična situacija, ali je sada jedan brid konveksne ljuske paralelan s pravcem, te su dva primjera iz jedne klase (crvene) i jedan primjer iz druge klase (plave) jednako udaljeni od pravca.

Dobro, sada kada znamo kako želimo postaviti hiperravninu (tako da maksimizira marginu), možemo se zapitati kako ćemo to postići. Kako možemo reći algoritmu strojnog učenja koju hiperravninu da *preferira*? Odgovor je: trebamo definirati **pristranosti preferencijom**. Kako definiramo pristranost preferencije kod algoritma strojnog učenja? Tako da tu pristranost ugradimo u empirijsku pogrešku i optimizacijski postupak. Pa, hajdemo onda definirati te komponente algoritma, tako da dobijemo upravo hipotezu s maksimalnom marginom.

Prisjetimo se najprije kako smo to radili do sada, kod poopćenih linearnih modela. Recept je bio sljedeći. Najprije smo definirali vjerojatnost oznaka skupa označenih primjera. To znači da smo pretpostavili da izlaz modela odgovara nekoj teorijskoj distribuciji (Gaussovoj, Bernoullijevoj, multinoullijevoj), koja je upravljana parametrima $\mathbf{w}$. Zatim smo napisali izraz za logaritam vjerojatnosti oznaka pod takvom distribucijom i tretirali ga kao funkciju parametara $\mathbf{w}$. Nakon toga izveli smo **empirijsku pogrešku** kao negativnu vrijednost tog logaritma. Konačno, nakon toga razvili smo **optimizacijski postupak**, već ovisno o tome je li pogreška imala rješenje u zatvorenoj formi ili nije.

Kod SVM-a, međutim, ići ćemo posve drugim putem. Krenut ćemo odmah od onoga što želimo dobiti – **maksimalne margine** – i direktno nju definirati kao optimizacijski problem. Onda će iz toga, praktički kao nusprodukt, ispasti funkcija pogreške i funkcija gubitka.

1.1 Formulacija optimizacijskog problema

Formalizirajmo sada problem maksimalne margine. U nastavku će nam biti lakše težinu w_0 tretirati odvojeno od drugih težina, kao što smo radili kada smo pričali o geometriji linearnog modela. Model SVM-a je najjednostavniji mogući:

$$h(\mathbf{x}; \mathbf{w}, w_0) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

Granica između klasa je hiperravnina $h(\mathbf{x}) = 0$. Kao i uvijek, to je $(n-1)$-dimenzijska hiperravnina ugrađena u n-dimenzijski prostor. Oznake primjera modela neka su $y \in \{-1, +1\}$; tako će biti lakše nego da radimo sa 0 i 1. 7

Predikcija modela ovisi o tome je li primjer na jednoj ili drugoj strani hiperravnine, što možemo detektirati na temelju predznaka. Dakle, predikcija oznake je: 8

$$y = \operatorname{sgn}\big(h(\mathbf{x})\big)$$

Uz našu pretpostavku da su primjeri linearno odvojivi, postoje težine $\mathbf{w}$ i w_0 takve da:

$$\forall (\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}.\ y^{(i)} h(\mathbf{x}^{(i)}) \geqslant 0$$

Ovo jednostavno slijedi iz pretpostavke da su primjeri linearno odvojivi. Naime, ako su primjeri linearno odvojivi, onda ih hipoteza $h(\mathbf{x})$ sve ispravno klasificira, a to znači da je $h(\mathbf{x}^{(i)}) = y^{(i)}$ za svaki označeni primjer $(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}$, a to je isto kao da smo napisali $y^{(i)} h(\mathbf{x}^{(i)}) \geqslant 0$ (ili je primjer pozitivan pa je i izlaz modela pozitivan ili nula, pa je umnožak pozitivan ili nula, ili je primjer negativan pa je izlaz modela negativan, pa je umnožak pozitivan; sjetite se da smo upravo umnožak $y^{(i)} h(\mathbf{x}^{(i)})$ koristili kod usporedbe funkcije gubitaka).

Koliko hipoteza postoji za koje ovo vrijedi? (Pretpostavite $\mathcal{X} \in \mathbb{R}^n$.) Odgovor je: ima ih beskonačno mnogo, odnosno postoji beskonačno mnogo $(w_0, \mathbf{w})$ za koje ovo vrijedi.

Sad uvodimo induktivnu pristranost preferencijom: nas zanima ona hipoteza koja daje rješenje **maksimalne margine**. Prisjetimo se: margina je udaljenost hiperravnine do najbližeg primjera. Dakle, želimo da hiperravnina prolazi tako da je najudaljenija od najbližih primjera dviju klasa. Zapravo, kad se prisjetimo slike s konveksnim ljuskama, to znači da je hiperravnina upravo jednako udaljena od najbližih primjera iz obaju klasa. Znamo da je **predznačena udaljenost** primjera od hiperravnine jednaka:

$$d = \frac{h(\mathbf{x})}{\|\mathbf{w}\|} = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + w_0}{\|\mathbf{w}\|}$$

Nas zanimaju samo hiperravnine koje ispravno sve klasificiraju primjere. To znači da vrijedi $y^{(i)}h(\mathbf{x}^{(i)}) \geqslant 0$, bio primjer pozitivan ili negativan. Dalje, to znači da je

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

nepredznačena udaljenost primjera $\mathbf{x}^{(i)}$ od hiperravnine, bio on pozitivan ili negativan (dakle, uvijek pozitivan broj, neovisno o oznaci primjera). Nadalje, po definiciji, margina je udaljenost hiperravnine do **najbližeg primjera**:

$$\min_i \left\{ \frac{y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + w_0)}{\|\mathbf{w}\|} \right\} = \frac{1}{\|\mathbf{w}\|} \min_i \left\{ y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + w_0) \right\}$$

gdje smo na desnoj strani izlučili $\|\mathbf{w}\|$ izvan funkcije min, budući da je norma vektora težina za sve primjere ista. Provedite ovdje nekoliko minuta dok se niste uvjerili da razumijete izraz koji smo upravo napisali. Funkcija min iterira po svim primjerima iz $\mathcal{D}$ i računa udaljenost primjera od hiperravnine, te nalazi najmanju takvu udaljenost, tj. udaljenost do najbližeg primjera, bilo pozitivnog ili negativnog. I sada, budući da želimo **maksimalnu marginu**, ovu udaljenost koju smo upravo napisali želimo maksimizirati, tj. tražimo takvu hiperravninu (takve parametre $\mathbf{w}$ i w_0) koji će maksimizirati tu udaljenost:

$$\underset{\mathbf{w},w_0}{\operatorname{argmax}} \left\{ \frac{1}{\|\mathbf{w}\|} \min_i \left\{ y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + w_0) \right\} \right\}$$

Opet, provedite ovdje nekoliko minuta dok se niste uvjerili da razumijete što smo ovdje napisali. Konceptualno, funkcija argmax iterira po svim mogućim hiperravninama, za svaku hiperravninu izračunavamo pomoću funkcije min udaljenost do najbližeg primjera, te odabiremo onu hiperravninu, tj. one parametre $\mathbf{w}$ i w_0, za koju je ta udaljenost najveća. Ako je $\mathcal{D}$ linearno odvojiv, a pretpostavili smo da jest, onda postoji samo jedna takva hiperravnina. Također, za tu hiperravninu postojat će barem dva primjera koji će od nje biti jednako udaljeni, i to jedan pozitivan s jedne strane hiperravnine te jedan negativan s druge strane hiperravnine. Naime, kada to ne bi bilo tako – kada bi jedan primjer iz jedne klase bio bliži hiperravnini a drugi iz druge klase malo dalji – onda ta hiperravnina ne bi bila rješenje maksimalne margine, jer bismo je uvijek mogli malo odmaknuti od bližeg primjera i približiti onom daljem primjeru i time maksimizirati marginu. Dakle, hiperravnina koja maksimizira marginu sigurno je pozicionirana tako da su njoj najbliži primjeri iz pozitivne i negativne klase od nje jednako udaljeni. Primijetite da sve ovo slijedi iz gornjeg izraza, tj. ne moramo više uvoditi nikakve dodatne uvjete.

Ovime smo definirali kakvu hiperravninu želimo, odnosno definirali smo optimizacijski problem. Nažalost, ovako definiran optimizacijski problem ne možemo izravno riješiti: poteškoća je u tome što imamo min-izraz unutar argmax-izraza. Umjesto toga, problem trebamo nekako preformulirati, tako da postane jednostavniji. Pokazuje se da to možemo napraviti, i to tako da se riješimo min funkcije. Ideja je da pretpostavimo da je za primjer koji je najbliži margini izlaz modela $\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$ jednak nekoj konstanti, pa onda uopće ne trebamo tražiti najbliži primjer. Kako to možemo napraviti? Pa, trik je u tome da se sjetimo da vektor težina $(w_0, \mathbf{w})$ možemo

proizvoljno skalirati, a da to neće utjecati na orijentaciju hiperravnine niti na udaljenosti između primjera od hiperravnine, ali će međutim to utjecati na izlaz hipoteze. Pretpostavimo onda da je vektor težina $(w_0, \mathbf{w})$ skaliran upravo tako da hipoteza za najbliži pozitivan primjer daje izlaz $+1$ te, posljedično, za najbliži negativan primjer daje izlaz -1. Drugim riječima, pretpostavimo da za primjer koji je najbliži hiperravnini vrijedi: 9

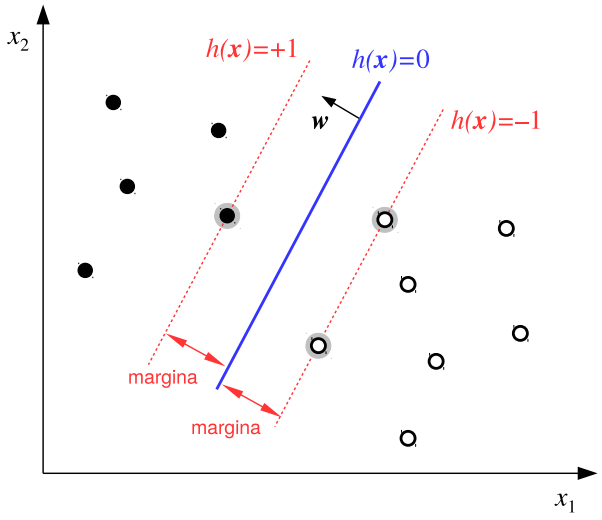
$$y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0) = 1$$

Ovo može vrijediti za više od jednog primjera. Zapravo, primijetite da će ovo sigurno vrijediti za barem dva primjera, po jedan sa svake strane hiperravnine, jer inače hiperravnina ne bi bila na poziciji maksimalne margine. 10

Gornji uvjet vrijedi, dakle, za primjere iz $\mathcal{D}$ koji su najbliži margini, a takvih je barem dva (po jedan sa svake strane), a može ih biti i više. Što je sa svim drugim primjerima u $\mathcal{D}$? Svi drugi primjeri bit će još udaljeniji od margine, dakle za njih će izlaz modela biti $h(\mathbf{x}) > 1$ ako su pozitivni odnosno $h(\mathbf{x}) < -1$ ako su negativni. Dakle, možemo reći da za sve primjere u $\mathcal{D}$ (i one najbliže hiperravninini i one koji to nisu), od $i = 1$ do $i = N$, vrijedi sljedeće:

$$y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + w_0) \geqslant 1, \qquad i = 1, \ldots, N$$

Prikažimo to grafički:



Na slici je prikazan dvodimenzijski ulazni prostor s primjerima klase $y = 1$ (puni kružići) i klase $y = 0$ (prazni kružići). Granica je pravac za koji $h(\mathbf{x}) = 0$. Pozitivni primjeri (oni za koje $y = 1$) su na pozitivnoj strani pravca, tj. strani u smjeru normale $\mathbf{w}$. Tri su primjera najbliža pravcu $h(\mathbf{x}) = 0$, i to jedan primjer iz klase $y = 1$ i dva primjera iz klase $y = 0$ (ta tri primjera označena su sivim rubom). Izlaz hipoteze za te primjere je $h(\mathbf{x}) = +1$ (za primjer iz klase $y = 1$) odnosno $h(\mathbf{x}) = -1$ (za dva primjera iz klase $y = 0$). Da je to tako slijedi iz našeg uvjeta da za primjere koji su najbliži pravcu (odnosno općenito hiperravnini) mora vrijediti $yh(\mathbf{x}) = 1$. Na slici vidimo i marginu, a to je udaljenost od pravca do najbližeg primjera s jedne i druge strane. Za tri primjera za koja vrijedi $yh(\mathbf{x}) = 1$ kažemo da se nalaze "na margini".

Nakon ovog zgodnog trika – da za primjere koji su najbliži hiperravnini pretpostavimo $yh(\mathbf{x}) = 1$ – optimizacijski se problem iz:

$$\underset{\mathbf{w},w_0}{\operatorname{argmax}} \left\{ \frac{1}{\|\mathbf{w}\|} \underbrace{\min_i \left\{ y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0) \right\}}_{=1} \right\}$$

svodi na:

$$\underset{\mathbf{w},w_0}{\operatorname{argmax}} \frac{1}{\|\mathbf{w}\|}$$

Međutim, ovo vrijedi samo ako vrijedi naša pretpostavka, stoga i nju moramo nekako ugraditi u optimizacijski postupak. To radimo tako da u optimizacijski postupak dodamo sljedeća **ograničenja**:

$$y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + w_0) \geqslant 1, \qquad i = 1, \ldots, N$$

Došli smo dakle, do toga da želimo maksimizirati $\frac{1}{\|\mathbf{w}\|}$ uz gornja ograničenja. Sada možemo primijetiti da je maksimizator od $\frac{1}{\|\mathbf{w}\|}$ ekvivalentan minimizatoru od $\|\mathbf{w}\|$, a taj je pak ekvivalentan minimizatoru od $\|\mathbf{w}\|^2$ (jer je L_2-norma konveksna i nenegativna funkcija). Još ćemo sve to pomnožiti s $\frac{1}{2}$ radi kasnije matematičke jednostavnosti. Konačna formulacija optimizacijskog problema maksimalne margine onda je sljedeća:

$$\underset{\mathbf{w}, w_0}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

uz ograničenja:

$$y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + w_0) \geqslant 1, \quad i = 1, \ldots, N$$

Pogledajmo što smo zapravo napravili. Krenuli smo od problema maksimalne margine, i napisali ga kao maksimizaciju udaljenosti između najbližih primjera. To nam je dalo izraz argmax-min, koji nije prikladan za optimizaciju. Onda smo izveli lukav trik: pretpostavili smo da za primjere koji su najbliži hiperravnini hipoteza daje ± 1. To nam je omogućilo da se riješimo izraza min. Konačno, uz malo algebre, došli smo do toga da želimo minimizirati $\frac{1}{2}\|\mathbf{w}\|^2$ uz ograničenja $y^{(i)}h(\mathbf{x}^{(i)}) \geqslant 1$. 11

Naš optimizacijski problem sveo se na ciljnu funkciju koju želimo minimizirati i ograničenja koja pritom moramo poštovati. Budući da je ciljna funkcija **konveksna**, ovo je tipičan problem koji se u teoriji optimizacije naziva **konveksna optimizacija uz ograničenja**, odnosno preciznije problem **kvadratnog programiranja**. 12

Očito, da bismo mogli dalje, trebamo pogledati o čemu se tu radi.

2 Optimizacija uz ograničenja

Općenito, **optimizacijski problem uz ograničenja** (engl. *constrained optimization problem*) definiran je na sljedeći način:

$$\begin{aligned} \text{minimizirati} \quad & f(\mathbf{x}) \\ \text{uz ograničenja} \quad & g_i(\mathbf{x}) \leqslant 0, \quad i = 1, \ldots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, p \end{aligned}$$

Pri čemu je funkcija $f : \mathbb{R}^n \to \mathbb{R}$ **ciljna funkcija** (engl. *objective function*) koju minimiziramo, $h_i : \mathbb{R}^n \to \mathbb{R}$ su **ograničenja jednakosti** (engl. *equality constraints*), a $g_i : \mathbb{R}^n \to \mathbb{R}$ su **ograničenja nejednakosti** (engl. *inequality constraints*). Primijetite da možemo imati više ograničenja jednakosti i više ograničenja nejednakosti (a možemo imati i samo jednu od te dvije vrste ograničenja).

Ovakav oblik optimizacijskog problema (minimizacija ciljne funkcije, ograničenja definirana tako da su manja ili jednaka nuli) naziva se **standardni oblik**. Standardni oblik možda se čini malo ograničavajućim, ali to zapravo nije. Naime, ako umjesto minimizacije želimo maksimizaciju funkcije $f(\mathbf{x})$, onda možemo jednostavno minimizirati funkciju $-f(\mathbf{x})$. Nadalje, sva ograničenja (ne)jednakosti daju se uvijek svesti na standardni oblik. Konkretno, ograničenje jednakosti $h(\mathbf{x}) = c$ možemo napisati kao $h(\mathbf{x}) - c = 0$, dok ograničenja nejednakosti $g(\mathbf{x}) \leqslant c$ ili $g(\mathbf{x}) \geqslant c$ možemo napisati kao $g(\mathbf{x}) - c \leqslant 0$ odnosno $c - g(\mathbf{x}) \leqslant 0$.

Kod optimizacije s ograničenjima tražimo minimum koji zadovoljava sva ograničenja. Točke koje zadovoljavaju ograničenja nazivamo **ostvarivim točkama** (engl. *feasible points*) ili **ostvarivim područjem** (engl. *feasibility region*).

Poseban slučaj gornjeg optimizacijskog problema jest onaj kod kojega je funkcija $f(\mathbf{x})$ konveksna. Tada govorimo o **konveksnome optimizacijskom problemu** (engl. *convex optimization problem*). Nadalje, poseban slučaj konveksnog optimizacijskog problema je **kvadratni program** (engl. *quadratic program, QP*), kod kojega je ciljna funkcija kvadratna (pa time i konveksna), a ograničenja su afine funkcije. Primijetite da je upravo takav optimizacijski problem maksimalne margine. Naime, ciljna funkcija je kvadratna: 13 14

$$\frac{1}{2}\|\mathbf{w}\|^2 = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

dok su ograničenja nejednakosti linearna:

$$y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + w_0) \geqslant 1, \quad i = 1, \ldots, N$$

Postoji niz metoda za rješavanje kvadratnog programa, npr. **metode kazne** (engl. *penalty methods*), **metode unutarnje točke** (engl. *interior point methods / barrier methods*), **koordinatni spust**, metoda **konjugiranog gradijenta** i **Lagrangeova dualnost**. Mi ćemo koristiti zadnje navedenu metodu, koja je, u kombinaciji s algoritmom **slijedne minimalne optimizacije** (engl. *sequential minimal optimization, SMO*), bila prva korištena za rješavanje problema maksimalne margine. Glavna ideja te metode jest da ćemo preći u tzv. **dualnu formulaciju optimizacijskog problema**. Uskoro ćemo vidjeti što to zapravo znači. Vidjet ćemo i koje su prednosti tog pristupa.

Pogledajmo, dakle, kako pristupiti kvadratnom programiranju preko Lagrangeove dualnosti.

3 Lagrangeova dualnost

Lagrangeova dualnost zasniva se na metodi **Langrangeovih multiplikatora**. Ideja metode Lagrangeovih multiplikatora jest da **preformuliramo** optimizacijski problem s ograničenjima (ne nužno konveksan!) tako da ta ograničenja eksplicitno **ugradimo** u ciljnu funkciju. Pogledajmo kako. 15

3.1 Lagrangeova funkcija

Početni problem je:

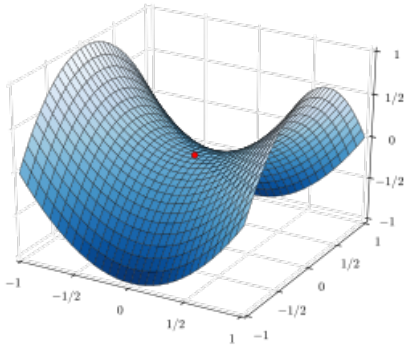
$$\begin{aligned} \text{minimizirati} \quad & f(\mathbf{x}) \\ \text{uz ograničenja} \quad & g_i(\mathbf{x}) \leqslant 0, \quad i = 1, \ldots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, p \end{aligned}$$

Ciljnu funkciju i ograničenja kombiniramo u novu funkciju:

$$L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i g_i(\mathbf{x}) + \sum_{i=1}^{p} \beta_i h_i(\mathbf{x})$$

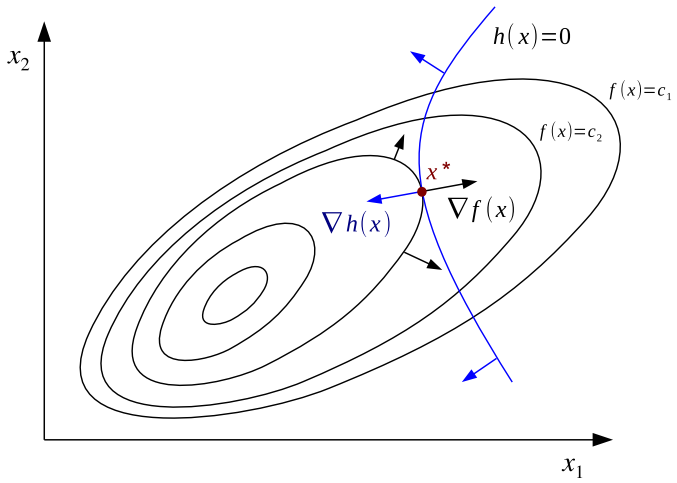
gdje $\alpha_i \geqslant 0$. Ovu funkciju nazivamo **Lagrangeova funkcija**. Vrijednosti α_i i β_i su **Lagrangeovi multiplikatori** (množitelji) za ograničenja nejednakosti odnosno ograničenja jednakosti.

Rješenje originalnog optimizacijskog problema s ograničenjem je točka u kojoj je gradijent Lagrangeove funkcije jednak nuli, $\nabla L = 0$, tj. **stacionarna točka** Lagrangeove funkcije. Pokazuje se da je ta stacionarna točka zapravo **sedlo (saddle point)** Lagrangeove funkcije i ta je točka minimum funkcije po $\mathbf{x}$ i maksimum funkcije po $\boldsymbol{\alpha}$ i $\boldsymbol{\beta}$. Ovako to izgleda:



Ovo je vrlo zanimljivo, i ovo je sedlo vrlo lijepo, no otkud sve to? Hajdemo ovo pogledati malo detaljnije. Lagrangeova funkcija kodira dvije vrste ograničenja: **ograničenja jednakosti** i **ograničenja nejednakosti**. Pogledajmo zasebno ta dva slučaja. Fokusirat ćemo se na slučaj kada je ciljna funkcija konveksna (premda Lagranegova metoda nije ograničena samo na konveksne funkcije; jedini zahtjev jest da su ciljna funkcija i funkcije ograničenja derivabilne). Bez smanjenja općenitosti, ograničit ćemo se na funkciju dvije varijable, kako bismo to mogli skicirati.

3.2 Ograničenja jednakosti



Na slici su prikazane izokonture ciljne funkcije $f(\mathbf{x})$. To je funkcija čiji minimum tražimo. Minimum se nalazi negdje u sredini najmanje konture. Funkcija $h(\mathbf{x}) = 0$ (plava krivulja) je ograničenje koje naše rješenje mora zadovoljiti. To znači da u obzir za rješenje dolaze samo one točke koje se nalaze na toj krivulji. Na našoj slici $h(\mathbf{x}) = 0$ je krivulja, no u općenitom slučaju, $h(\mathbf{x}) = 0$ definira n-dimenzijsku površinu ugrađenu u prostor $\mathbb{R}^n \times \mathbb{R}$. U svakoj točki na površini $h(\mathbf{x}) = 0$, gradijent $\nabla h(\mathbf{x})$ bit će okomit na tu površinu (to mora biti po definiciji gradijenta). Neka je točka $\mathbf{x}^*$ točka na površini ograničenja $h(\mathbf{x}) = 0$ koja minimizira $f(\mathbf{x})$ (označena crveno). To je točka u kojoj smo se uspjeli najviše približiti globalnome minimumu funkcije $f(\mathbf{x})$, ali smo ipak ostali na površini $h(\mathbf{x}) = 0$ koja definira ostvarive točke. (Kao muha koja kroz staklo želi doći što blizu pekmezu; doći će do točke na staklu gdje je najbliža pekmezu, ali je i dalje na staklu).

Sad primijetite da vektor $\nabla f(\mathbf{x}^*)$ također mora biti okomit na povšinu $h(\mathbf{x}) = 0$. U protivnom bismo se naime uvijek mogli pomaknuti po površini ograničenja tako da se vrijednost $f(\mathbf{x})$ smanji. Tek ako je gradijent okomit, znači da smo na minimalnoj mogućoj točki.

Temeljem ovih opažanja, zaključujemo da u točki $\mathbf{x}^*$ vektori gradijenta ∇f i ∇h moraju biti **kolinearni (paralelni ili antiparalelni)**. To onda znači da u točki $\mathbf{x}^*$ mora postojati konstanta β

za koju vrijedi:

$$\nabla f(\mathbf{x}^*) + \beta \nabla h(\mathbf{x}^*) = 0$$

tj. točka u kojoj linearna kombinacija vektora ∇f i ∇h iščezava. Točka $\mathbf{x}^*$ za koju ovo vrijedi upravo je rješenje našeg početnog optimizacijskog problema s ograničenjem!

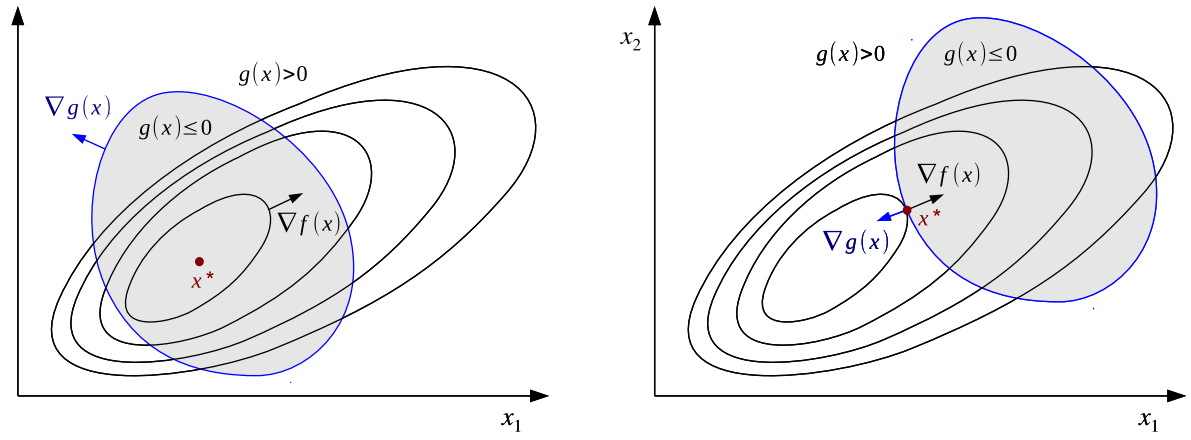
Sada pretpostavimo da ova jednadžba odgovara stacionarnoj točki neke funkcije. Ako je tako, onda bi ta funkcija bila ova:

$$L(\mathbf{x}, \beta) \equiv f(\mathbf{x}) + \beta h(\mathbf{x})$$

i to je upravo naša Lagrangeova funkcija! Pa zaključujemo: stacionarna točka Lagrangeove funkcije je rješenje našeg problema s ograničenjem.

3.3 Ograničenja nejednakosti

Pogledajmo sada slučaj s **ograničenjima nejednakosti**, jer to je zapravo ono što će nam trebati za naš problem maksimalne margine. Tu su moguća dva slučaja: minimum funkcije $f(\mathbf{x})$ nalazi se unutar ostvarivog područja ili se nalazi izvan.



Na lijevoj slici prikazana je situacija kada je minimum *unutar* ostvarenog područja. Tada kažemo da ograničenje **nije aktivno**, te funkcija $g(\mathbf{x})$ ne igra nikakvu ulogu.

Na desnoj slici prikazana je situacija kada je minimum *izvan* ostvarivog područja. Tada kažemo da je ograničenje **aktivno** i točka minimuma $\mathbf{x}^*$ nalazi se na površini $g(\mathbf{x}) = 0$, što je bliže moguće neograničenom minimumu. To nadalje znači da sve točke $\mathbf{x}$ u ostvarivom području imaju vrijednost $f(\mathbf{x})$ veću od minimuma, pa gradijent $\nabla f(\mathbf{x})$ pokazuje prema ostvarivom području, dok $\nabla g(\mathbf{x})$ pokazuje od njega. Posljedično, $\nabla f(\mathbf{x})$ i $\nabla g(\mathbf{x})$ su antiparalelni vektori te vrijedi:

$$\nabla f(\mathbf{x}^*) = -\alpha \nabla g(\mathbf{x}^*)$$

za neku konstantu $\alpha > 0$.

Prema tome, uzevši u obzir oba ova slučaja, minimizaciji $f(\mathbf{x})$ uz uvjet $g(\mathbf{x})$ odgovara nalaženje **stacionarne točke** sljedeće Lagrangeove funkcije:

$$L(\mathbf{x}, \alpha) \equiv f(\mathbf{x}) + \alpha g(\mathbf{x})$$

uz $\alpha \geqslant 0$. Ako $\alpha = 0$, onda ograničenje $g(\mathbf{x})$ nije aktivno.

Primijetite da za točku ostvarivog minimuma $\mathbf{x}^*$ mora vrijediti ili $\alpha = 0$ (neaktivno ograničenje) ili $g(\mathbf{x}) = 0$ (aktivno i ispoštovano ograničenje). To sažeto možemo napisati kao:

$$\alpha g(\mathbf{x}) = 0$$

Ovaj se uvjet naziva **komplementarna labavost** (engl. *complementary slackness*).

Početna ograničenja jednakosti i nejednakosti, zajedno s ova dva uvjeta koja smo upravo izveli ($\alpha \geqslant 0$ i $\alpha g(\mathbf{x}) = 0$), čine tzv. **Karush-Kuhn-Tuckerove (KKT) uvjete**. To su uvjeti koji nužno vrijede u točki rješenja. 16

Tako smo, dakle, dobili Lagrangeovu funkciju i pripadne uvjete KKT. Evo sada sve na jednom mjestu:

▸ **Lagrangeova funkcija i uvjeti KKT**

$$\begin{aligned} \text{minimizirati} \quad & f(\mathbf{x}) \\ \text{uz ograničenja} \quad & g_i(\mathbf{x}) \leqslant 0, \quad i = 1, \dots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \end{aligned}$$

Lagrangeova funkcija:

$$L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i g_i(\mathbf{x}) + \sum_{i=1}^{p} \beta_i h_i(\mathbf{x})$$

U stacionarnoj točki vrijede uvjeti KKT:

$$\begin{aligned} g_i(\mathbf{x}) &\leqslant 0, \quad i = 1, \dots, m \\ h_i(\mathbf{x}) &= 0, \quad i = 1, \dots, p \\ \alpha_i &\geqslant 0, \quad i = 1, \dots, m \\ \alpha_i g_i(\mathbf{x}) &= 0, \quad i = 1, \dots, m \end{aligned}$$

Primijetite da Lagrangeove funkcije za ograničenje jednakosti i nejednakosti možemo kombinirati u jednu funkciju s oba ograničenja. Također, ako imamo više ograničenja, samo ih sve pridodamo u Lagrangeovu funkciju.

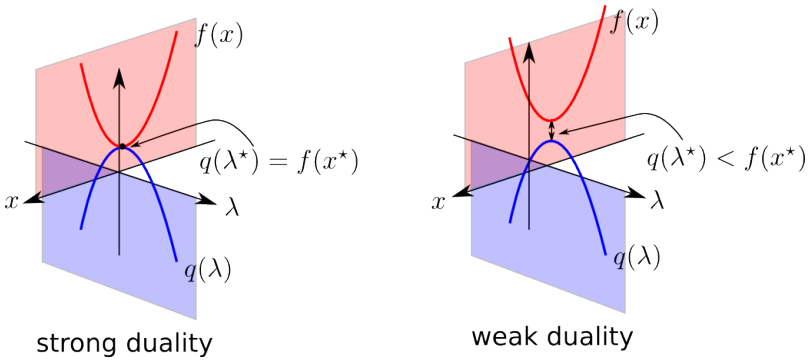
Sada smo se (nadam se) uvjerili da Lagrangeova funkcija i uvjeti KKT imaju smisla. No, kako nalazimo stacionarnu točku Lagrangeove funkcije? To nas dovodi do tzv. **načela dualnosti**.

3.4 Načelo dualnosti

U teoriji optimizacije postoji tzv. **načelo dualnosti** (engl. *duality principle*), koje nam kaže da optimizacijski problem možemo sagledavati na dva načina:

- **Primarni problem** (engl. *primal problem*): minimizacija funkcije $f(\mathbf{x})$
- **Dualni problem** (engl. *dual problem*): nalaženje **donje ograde** primarnog problema

Odnos između primarnog i dualnog optimizacijskog problema možemo prikazati ovako:



Slike prikazuju primarni problem (crvena krivulja) i dualni problem (plava krivulja). Primarni problem i dualni problem imaju svaki svoje varijable po kojima optimiziramo: primarni problem optimiziramo po primarnim varijablama (na slici je to varijabla x), a dualni problem po dualnim varijablama (na slici je to varijabla λ). Primarne varijable i dualne varijable su različite varijable, pa je na slici prikazano tako da su dimenzije primarnog problema i dualnog problema različite (primarni problem: crvena ravnina, dualni problem: plava ravnina). Plava krivulja odgovara dualnom problemu, i ona je donja ograda primarnog problema (dakle, minimum funkcije $f(\mathbf{x})$ veći je ili jednak vrijednosti plave krivulje). Općenito, rješenja primarnog i dualnog problema se ne moraju poklapati već može postojati tzv. **procjep dualnosti** (engl. *duality gap*). To je prikazano na desnoj slici. Uz određene uvjete, kod konveksne optimizacije dualni procjep 17
jednak je nuli. Tada govorimo o **jakoj dualnost** (engl. *strong duality*) (lijeva slika). Rješenje se onda nalazi u točki sedla: minimum po primarnim varijablama (crvena krivulja) a maksimum po dualnim variablama (plava krivulja).

U našem slučaju vrijedi jaka dualnost, budući da je problem maksimalne margine konveksan optimizacijski problem (dapače, to je kvadratni program). To znači da umjesto primarnog problema možemo rješavati dualni problem, i da ćemo dobiti identično rješenje. Ovdje se odmah nameće pitanje zašto bismo to uopće htjeli? Zašto rješavati dualni problem umjesto primarnog, kada ćemo ionako dobiti isto rješenje? Odgovor na to pitanje ostavit ćemo za kasnije. Ovdje 18
ćemo samo reći da rješavanje problema maksimalne margine u dualu ima niz važnih praktičnih prednosti koje su vrlo primamljive.

Pogledajmo kako načelo dualnosti izgleda kod Lagrangeove funkcije – to je **Lagrangeova dualnost**. Lagrangeova funkcija $L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ je funkcija **primarnih varijabli** $\mathbf{x}$ i **dualnih varijabli** $\boldsymbol{\alpha}$ i $\boldsymbol{\beta}$. Već znamo da je rješenje optimizacijskog problema stacionarna točka Lagrangeove funkcije, tj. točka za koju vrijedi:

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = 0$$

Međutim, rješenje ove jednadžbe ne mora dovesti do uklanjanja dualnih varijabli. Općenito, dobit ćemo rješenje koje, za neke $\boldsymbol{\alpha}$ i $\boldsymbol{\beta}$, minimizira Lagrangeovu funkciju L po primarnim varijablama $\mathbf{x}$. Drugim riječima, dobit ćemo funkciju:

$$\tilde{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Ova se funkcija naziva **dualna Lagrangeova funkcija** (primijetite tildu!). Provedite ovdje neko vrijeme da shvatite što ova funkcija zapravo radi. Za fiksirani $\boldsymbol{\alpha}$ i $\boldsymbol{\beta}$ (dualne varijable), funkcija vraća minimum Lagrangeove funkcije po primarnoj varijabli $\mathbf{x}$. Zbog načela dualnosti, ova funkcija je **donja ograda** primarnog problema: to znači da je minimum od $L(\mathbf{x}, \alpha, \beta)$ veći ili jednak vrijednostima $\tilde{L}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ za svaki $\boldsymbol{\alpha}$ i $\boldsymbol{\beta}$. Nadalje, zato što vrijedi jaka dualnost, minimum primarnog problema možemo pronaći tako da *maksimiziramo* dualni problem, je će nas to dovesti u točku sedla: minimum po primarnim parametrima, a maksimum po dualnim parametrima (v. prethodnu sliku). Drugim riječima, trebamo riješiti sljedeći konveksni optimizacijski problem:

$$\begin{aligned} &\text{maksimizirati} && \tilde{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &\text{uz ograničenja} && \alpha_i \geqslant 0, \quad i = 1, \ldots, m \end{aligned}$$

Zaključujemo: **minimizacija ciljne funkcije** istovjetna je **maksimizaciji dualne funkcije** (ako vrijedi jaka dualnost, što kod nas jest slučaj!).

Sažmimo sve ovo na jednom mjestu. Želimo riješiti sljedeći optimizacijski problem:

▸ **Konveksna optimizacija u dualu**

$$\begin{aligned} \text{minimizirati} \quad & f(\mathbf{x}) \\ \text{uz ograničenja} \quad & g_i(\mathbf{x}) \leqslant 0, \quad i = 1, \ldots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, p \end{aligned}$$

Pripadna Lagrangeova funkcija je:

$$L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i g_i(\mathbf{x}) + \sum_{i=1}^{p} \beta_i h_i(\mathbf{x})$$

Dualna Lagrangeova funkcija je:

$$\tilde{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Dualni problem je:

$$\begin{aligned} \text{maksimizirati} \quad & \tilde{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ \text{uz ograničenja} \quad & \alpha_i \geqslant 0, \quad i = 1, \ldots, m \end{aligned}$$

Ako je problem konveksan, vrijedi jaka dualnost, pa je rješenje dualnog problema ujedno i rješenje primarnog problema.

A sada kada sve ovo znamo, vratimo se našem poslu – optimizaciji maksimalne margine...

4 Optimizacija maksimalne margine

Dakle, naš kvadratni program maksimalne margine riješit ćemo tako da ćemo ograničenja uvesti u Lagrangeovu funkciju, koja je funkcija originalnih (primarnih) varijabli i novih (dualnih) varijabli. Zatim ćemo iskazati minimum Lagrangeove funkcije po primarnim varijablama, što će nam dati dualnu Lagrangeovu funkciju. Naposlijetku ćemo maksimizirati tu funkcije, jer će nam njezina maksimizacija dati minimum primarne funkcije, budući da se radi o konveksnom problemu. Brilijantno!

Pa, krenimo. Prisjetimo se, optimizacijski problem maksimalne margine bio je:

$$\underset{\mathbf{w}, w_0}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

uz uvjete:

$$y^{(i)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + w_0) \geqslant 1, \quad i = 1, \ldots, N$$

Primijetite da imamo onoliko ograničenja koliko imamo primjera. Napišimo Lagrangeovu funkciju:

$$L(\mathbf{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i \Big\{ y^{(i)}\big(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} + w_0\big) - 1 \Big\}$$

gdje je $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$, $\alpha_i \geqslant 0$, vektor Lagrangeovih multiplikatora, po jedan za svaki primjer. Lagrangeova funkcija L je funkcija primarnih varijabli $\mathbf{w}$ i w_0 te dualnih varijabli $\boldsymbol{\alpha}$.

Sada prelazimo na dualnu formulaciju: po definiciji, dualna Lagrangeova funkcija je ona koja minimizira Lagrangeovu funkciju po primarnim varijablama. Dakle:

$$\tilde{L}(\boldsymbol{\alpha}) = \min_{\mathbf{w}, w_0} L(\mathbf{w}, w_0, \boldsymbol{\alpha})$$

Dobra stvar je da u ovom slučaju minimizacija Lagrangeove funkcije po primarnim varijablama ima rješenje u zatvorenoj formi. Minimizator $(\mathbf{w}^*, w_0^*)$ dobivamo deriviranje po $\mathbf{w}$ odnosno w_0 i izjednačavanje s nulom:

$$\frac{\partial L(\mathbf{w}, w_0, \boldsymbol{\alpha})}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \alpha_i y^{(i)} \mathbf{x}^{(i)}$$

$$\frac{\partial L(\mathbf{w}, w_0, \boldsymbol{\alpha})}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i y^{(i)} = 0$$

Posebno upamtite ovu prvu jednakost za $\mathbf{w}$, koji nam je bitna je povezuje primarne varijable $\mathbf{w}$ s dualnim varijablama $\boldsymbol{\alpha}$.

Ove dvije jednakosti sada uvrštavamo nazad u Lagrangeovu funkciju, kako bismo dobili njezin minimum, odnosno **dualnu Lagrangeovu funkciju**:

$$\begin{aligned}
\tilde{L}(\boldsymbol{\alpha}) &= \min_{\mathbf{w}, w_0} \Big(\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i \big\{ y^{(i)} \big(\mathbf{w}^{\mathrm{T}} \mathbf{x}^{(i)} + w_0 \big) - 1 \big\} \Big) \\
&= \min_{\mathbf{w}, w_0} \Big(\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i y^{(i)} \mathbf{w}^{\mathrm{T}} \mathbf{x}^{(i)} \underbrace{- w_0 \sum_{i=1}^{N} \alpha_i y^{(i)}}_{=0} + \sum_{i=1}^{N} \alpha_i \Big) \\
&= \frac{1}{2} \sum_{i=1}^{N} \alpha_i y^{(i)} (\mathbf{x}^{(i)})^{\mathrm{T}} \sum_{j=1}^{N} \alpha_j y^{(j)} \mathbf{x}^{(j)} - \sum_{i=1}^{N} \alpha_i y^{(i)} (\mathbf{x}^{(i)})^{\mathrm{T}} \sum_{j=1}^{N} \alpha_j y^{(j)} \mathbf{x}^{(j)} + \sum_{i=1}^{N} \alpha_i \\
&= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^{\mathrm{T}} \mathbf{x}^{(j)}
\end{aligned}$$

Ovaj izvod ne izgleda privlačno, ali je barem rezultat relativno pristojan. Uvjerite se da možete napraviti ovaj izvod.

Pogledajmo sada u kompletu kako izgleda **dualni optimizacijski problem SVM-a:**

▸ **Dualni optimizacijski problem SVM-a**

Maksimizirati:

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^{\mathrm{T}} \mathbf{x}^{(j)}$$

uz ograničenja

$$\alpha_i \geqslant 0, \quad i = 1, \ldots, N$$

$$\sum_{i=1}^{N} \alpha_i y^{(i)} = 0$$

U točki rješenja vrijede uvjeti KKT:

$$\begin{aligned}
y^{(i)} (\mathbf{w}^{\mathrm{T}} \mathbf{x}^{(i)} + w_0) &\geqslant 1, \quad i = 1, \ldots, N \\
\alpha_i &\geqslant 0, \quad i = 1, \ldots, N \\
\alpha_i \big(y^{(i)} h(\mathbf{x}^{(i)}) - 1 \big) &= 0, \quad i = 1, \ldots, N
\end{aligned}$$

Izvedenu jednakost $\sum_{i=1}^{N} \alpha_i y^{(i)} = 0$ morali smo dodati kao ograničenje jer se varijable α_i nismo riješili u dualnoj Lagrangeovoj funkciji (za razliku od primarne varijable $\mathbf{w}$, koje smo se riješili, pa prvu izvedenu jednakost više ne trebamo kao ograničenje.)

Ovo je i dalje problem **kvadratnog programiranja**, s time da, za razliku od primarnog problema, pored ograničenja nejednakosti, imamo i ograničenja jednakosti. Tako gledano, ispada da prijelazom iz primarne u dualnu formulaciju nismo baš puno dobili, naprotiv, imamo ograničenja jednakosti koja prije nismo imali. Međutim, kako smo već spomenuli, dualna formulacija ima niz prednosti. Jedna od njih je da se na ovaj kvadratni problem sad može primijeniti algoritam **slijedne minimalne optimizacije (SMO)**. Taj algoritam upravo iskorištava ograničenja jednakosti kako bi problem razbio na potprobleme, koje onda rješava analitički, te je zbog toga učinkovitiji od algoritama za općenite kvadratne programe. Nećemo ići u detalje; trebate samo znati da taj algoritam postoji i da se primjenjuje u dualnoj formulaciji. 19

Međutim, važan detalj koji ovdje trebamo uočiti jest da je, prelaskom iz primarnog u dualni problem, došlo do promjene u varijablama: primarni problem imao je $n+1$ primarnih varijabli, dok dualni ima N dualnih varijabli. Da li se ovo računalno isplati? Da, ako $N \ll n$, tj. **ako je broj primjera mnogo manji od broja značajki**. Tipične domene gdje je to redovito slučaj su bioinformatika, analiza teksta i analiza slike. Općenito, složenost kvadratnog programiranja od N varijabli je $\mathcal{O}(N^3)$. Međutim, SMO je prilagođen upravo ovom problemu i ima složenost $\mathcal{O}(N^2)$.

5 Dualni model SVM-a

Vratimo se sada na početak današnjeg prevanja. Na početku smo definirali **model SVM-a**, i to je bio linearan model:

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

Međutim, to je bila primarna formulacija modela. U dualnoj formulaciji više nemamo primarne parametre, tj. nemamo više težine. Umjesto toga imamo dualne parametre $\boldsymbol{\alpha}$. Prijelaz s primarnog na dualni model nam omogućava jednadžba koju smo već bili izveli:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y^{(i)} \mathbf{x}^{(i)}$$

Uvrštavanjem ovoga u izraz za primarni model, dobivamo:

$$h(\mathbf{x}) = \underbrace{\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0}_{\text{Primarno}} = \underbrace{\sum_{i=1}^{N} \alpha_i y^{(i)} \mathbf{x}^{\mathrm{T}}\mathbf{x}^{(i)} + w_0}_{\text{Dualno}}$$

Kako funkcionira predikcija u dualnom modelu? Da bismo klasificirali primjer $\mathbf{x}$, računamo **skalarni produkt** između $\mathbf{x}$ i svih primjera $\mathbf{x}^{(i)}$ iz skupa $\mathcal{D}$, pomnožen s težinom α_i i predznakom $y^{(i)}$. Računanje skalarnog produkta $\mathbf{x}^{\mathrm{T}}\mathbf{x}^{(i)}$ je zapravo računanje **sličnosti** između vektora $\mathbf{x}$ i $\mathbf{x}^{(i)}$, budući da:

$$\mathbf{x}^{\mathrm{T}}\mathbf{y} = \sum_{i=1}^{n} x_i y_i$$

Ovaj umnožak će biti to veći što se vektori podudaraju u više komponenenata, a to znači da su vektori to sličniji.

Dakle, umjesto da pohranjujemo težine $\mathbf{w}$, kod dualnog modela trebamo pohraniti primjere i njihove oznake. Umjesto:

$$h(\mathbf{x}; w_0, \mathbf{w})$$

imamo:

$$h(\mathbf{x}; \boldsymbol{\alpha}, \mathcal{D})$$

Pritom težinu w_0 ne trebamo, jer se ona može izračunati naknadno iz parametara α i primjera za učenje $\mathcal{D}$. 20

Objasnimo sada napokon zašto se ovaj algoritam zove **stroj potpornih vektora**. Iz uvjeta komplementarne labavosti:

$$\alpha_i \big(y^{(i)} h(\mathbf{x}^{(i)}) - 1 \big) = 0$$

slijedi da za svaki primjer $\mathbf{x}^{(i)}$ iz $\mathcal{D}$ vrijedi $\alpha_i = 0$ ili $y^{(i)} h(\mathbf{x}^{(i)}) = 1$. U prvom slučaju, ako $\alpha_i = 0$, onda taj vektor ne figurira u izračunu funkcije h. U drugom slučaju, ako $y^{(i)} h(\mathbf{x}^{(i)}) = 1$, onda to znači da primjer leži točno na margini (prisjetite se, $y^{(i)} h(\mathbf{x}^{(i)}) = 1$ je upravo bio uvjet koji smo definirali za primjere koji su najbliži hiperravnini, i to su primjeri koji leže na margini). Prema tome, iz ova dva slučaja slijedi da se u dualnoj formulaciji kao pribrojnici pojavljuju samo vektori koji leže točno ma margini. Te vektore nazivamo **potporni vektori** (engl. *support vectors*). Svi ostali vektori za koje $\alpha_i = 0$ uopće ne utječu na izlaz modela i možemo ih zanemariti kada radimo predikciju.

Alternativni pogled na ovo jest da je hiperravnina (u primarnom problemu) definirana linearnom kombinacijom potpornih vektora (u dualu). Ovo je vrlo bitno. To znači da će SVM model zapravo biti vrlo učinkovit kod predikcije. U mnogo slučajeva, umjesto da pohranjujemo vrlo mnogo težina $\mathbf{w}$, bit će dovoljno pohraniti manji broj potpornih vektora i pripadnih parametara $\boldsymbol{\alpha}$. Efektivno ćemo dobiti **rijetki model**.

Sažetak

- Stroj potpornih vektora je **linearan model** s **maksimalnom marginom** između primjera dviju klasa, što daje **dobru generalizaciju**
- Problem se svodi na **konveksnu optimizaciju s ograničenjima (kvadratno programiranje)**
- Primjenom **Lagrangeove dualnosti** problem se iz **primarne formulacije** može prebaciti u **dualnu formulaciju**
- Dualna optimizacijski problem učinkovito je rješiv algoritmom **SMO**
- Predikcija se radi na temelju **usporedbe** ulaznog primjera i odabranih označenih primjera, tzv. **potpornih vektora**

Bilješke

1 Pokušajmo odmah razriješiti misterij ovog čudnog naziva: zašto **stroj potpornih vektora**? Prvo, zašto "stroj". To nije skroz jasno. SVM tu nije usamljen, ima još strojeva u strojnom učenju, npr. Boltzmannov stroj (nećemo raditi), jezgreni stroj (radit ćemo), Helmholtzov stroj (nećemo raditi). Ovdje su dva objašnjenja: https://stats.stackexchange.com/q/261041/93766. Prema prvom, "stroj" dolazi jednostavno od "strojnog učenja", a prema drugom od povijesne povezanosti algoritama i strojeva, jer su u ranim danima računarske znanosti mnogi algoritmi doista bili fizički implementirani kao specijalizirani strojevi (slično bi se moglo reći i za "Turingov stroj"). Ovo se pitanje nedavno raspravljalo i na Twitteru: https://twitter.com/sam_power_825/status/1314921722630504448 (hvala Domagoju na informaciji). Što se drugog dijela naziva tiče, "potporni vektori", to je lako objasniti jednom kada se shvati kako algoritam radi, i to ćemo napraviti danas. No, objasnimo ovdje za nestrpljive barem ukratko ideju. SVM je linearan klasifikacijski model koji dakle primjere dviju klasa razdvaja hiperravninom. Tu hiperravninu možemo definirati vektorom težina $\mathbf{w}$, kao što to radi npr. logistička regresija, ali se, alternativno, hiperravnina može definirati kao linearna kombinacija primjera $\mathbf{x}$ iz skupa označenih primjera $\mathcal{D}$. Pritom nam ne trebaju svi primjeri iz $\mathcal{D}$, nego

samo neki. Ti neki primjeri koji nam trebaju zapravo su potpornji za tu hiperravninu, pa ih zato nazivamo **potporni vektori**.

2 SVM su osmisili ruski matematičari Vladimir Vapnik i Alexey Chervonenkis, i to još 1963. godine na "Instititutu za upravljačku znanost" (Institut Problem Upravleniya) u Moskvi. Vapnik je potom u devedestima emigrirao u SAD i radio za AT&T Bell Labs u New Jerseyu, gdje je, u suradnji s kolegicom Isabelle Guyon i kolegom Bernardom Boserom (u međuvremenu u braku), 1992. godine nadogradio ideju SVM-a s tzv. **jezgrenim funkcijama** (Boser et al., 1992). Jezgrene funkcije, o kojima ćemo mi pričati kroz dva tjedna, implicitan su način definiranja funkcije preslikavanja u prostor značajki više dimenzije, što je omogućilo primjenu SVM-a na nelinearne probleme. Daljnje proširenje bila je formulacija SVM-a s **mekom marginom** (Cortes and Vapnik, 1995), koju je Vapnik objavio zajedno s danskom znanstvenicom Corinnom Cortes 1995. godine, a koja je u to doba također radila u AT&T Bell Labs. SVM s mekom marginom omogućio je primjenu na (ne)linearno neodvojive probleme (probleme koji su linearno neodvojivi i probleme koji ostaju neodvojivi nakon preslikavanja u prostor značajki), i to je danas standardna formulacija SVM-a, koju ćemo i mi raditi. Inače, Vapnik od 2014. godine radi u Facebookovom istraživačkom centru FAIR, zajedno sa jakim imenima dubokog učenja, dok je Chervonenkis nažalost prije nekoliko godina tragično skončao u močvari u okolici Moskve.

3 Dokaz da maksimalna margina smanjuje pogrešku generalizacije temelji se na dokazivanju gornje ograde na složenost modela karakterizane pomoću tzv. **Vapnik-Chervonenkisove dimenzije (VC-dimenzije)**. VC-dimenziju osmislili su 1974. godine upravo Vladimir Vapnik i Alexey Chervonenkis, tvorci izvorne ideje SVM-a. Pojednostavljeno, VC-dimenzija je mjera složenosti modela definirana kao najveći broj primjera koje binarni klasifikator može točno razdvojiti, i to neovisno o oznakama tih primjera (tj. za sva moguća označavanja). Npr., VC-dimenzija pravca u dvodimenzijskome ulaznom prostoru je 3, jer pravac može razdvojiti najviše tri primjera u dvodimenzijskom prostoru, ako u obzir uzmemo sve moguće oznake tih primjer (kojih ukupno ima $2^3 = 8$). Ako dodamo četvrti primjer, onda imamo ukupno 16 mogućih označavanja, od kojih međutim 2 ne možemo razdvojiti pravcem (notorni "XOR" problem), što znači da VC-dimenzija pravca nije veća od 3. Što je VC-dimenzija veća, to je model veće složenosti. Više o VC-dimenziji možete pročitati u poglavlju 2.2 u skripti, a mnogo detaljnije u izvrsnoj knjizi (Shalev-Shwartz and Ben-David, 2014) (poglavlje 6). Ideja dokaza da maksimalna margina SVM-a daje najbolju generalizaciju temelji se na tome da se pokaže da maksimalna margina ograničava VC-dimenziju SVM-a, što znači da ograničava njegovu složenost, a to znači da sprječava prenaučenost i time poboljšava generalizaciju. Detalji ovoga vrlo su lijepo i didaktično pojašnjeni u (Mount, 2015) (dostupno ovdje: `https://winvector.github.io/margin/margin.pdf`). U (Shalev-Shwartz and Ben-David, 2014) (dio 26.3) možete naći dokaz koji se temelji na alternativnoj karakterizaciji složenosti modela pomoću **Rademacherove složenosti**.

4 **Teorem o odvajanju hiperravninom** (engl. *hyperplane separation theorem*) glasi ovako: Neka su A i B dva disjunktna neprazna konveksna podskupa u $\mathbb{R}^n$. Onda postoji ne-nul vektor $\mathbf{w}$ i realan broj c takav da:

$$\forall \mathbf{x}^A \in A, \forall \mathbf{x}^B \in B.\ (\mathbf{w}^{\mathrm{T}}\mathbf{x}^A \geqslant c) \wedge (\mathbf{w}^{\mathrm{T}}\mathbf{x}^B \leqslant c)$$

Pritom je $\mathbf{w}^{\mathrm{T}}\mathbf{x} = c$, gdje je $\mathbf{w}$ vektor normale, odvajajuća hiperravnina. U strojnom učenju, mi to uvijek formuliramo tako da je $c = 0$, tj. $\mathbf{w}^{\mathrm{T}}\mathbf{x} = 0$. Dokaz i više detalja možete pronaći na `https://en.wikipedia.org/wiki/Hyperplane_separation_theorem` odnosno u poglavlju 2.5 u (Boyd et al., 2004)
Inače, autor teorema o odvajanju hiperravninom je poljsko-njemački matematičar Hermann Minkowski. Minkowski je inače najpoznatiji po svojoj geometrijskoj interpretaciji teorije relativnosti Alberta Einsteina (kojemu je bio profesor matematike dok je ovaj studirao u Zürichu, na današnjem ETH Zürich) iz 1907. godine u smislu četverodimenzijskog prostora u kojem se prostor i vrijeme isprepliću, tzv. prostorvrijeme ili prostorno-vremenski kontinuum (*njem. Minkowski-Raum*), a koju je Einstein potom ugradio u svoju opću teoriju relativnosti. U računarstvu, pak, Minkowski nam je više poznat po **Mikowskijevoj udaljenosti**, koja je generalizacija euklidske udaljenosti i Manhattan-udaljenosti:

$$D(\mathbf{x}, \mathbf{y}) = \Big(\sum_i |x_i - y_i|^p \Big)^{\frac{1}{p}}$$

gdje za $p = 2$ i $p = 1$ dobivamo euklidsku udaljenost odnosno Manhattan-udaljenost. To je očigledno povezano s p-normom, o kojoj smo pričali u kontekstu regularizacije, i to u smislu da je Minkowskijeva

udaljenost između dvije točke jednaka p-normi razlike radijvektora tih točaka:

$$D(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p$$

Drugim riječima, regularizaciju p-normom vektora težina $\mathbf{w}$ mogli smo definirati i kao Mikowskijevu udaljenost vektora težina $\mathbf{w}$ od nul-vektora.

5 Definicije **konveksnog skupa** prisjetili smo se na pretprošlom predavanju u kontekstu definicije konveksne funkcije (domena konveksne funkcije mora biti konveksni skup). Prisjetimo se ovdje opet definicije konveksnog skupa. Skup A je konveksan ako i samo ako

$$\forall \mathbf{x}_1, \ldots, \mathbf{x}_n \in A, \forall \alpha_1, \ldots \alpha_n . \sum_i \alpha_1 = 1 \quad \text{vrijedi} \quad \sum_{i=1}^{n} \alpha_i \mathbf{x}_i \in A$$

tj. svaka linearna kombinacija točaka iz A je i sama unutar A.

6 Formalno (Boyd et al. (2004), poglavlje 2.1.4), **konveksna ljuska** skupa C, označena s conv C, je skup svih konveksnih kombinacija točaka iz C:

$$\text{conv } C = \left\{ \alpha_1 x_1 + \cdots + \alpha_k x_k \mid x_i \in C, \alpha_i \geqslant 0, i = 1, \ldots, k, \ \sum_i \alpha_i = 1 \right\}$$

7 Ili n-dimenzijska ravnina ugrađena u $(n+1)$-dimenzijski prostor, ako uvedemo "dummy" značajku $x_0 = 1$ i težinu w_0 uključimo u vektor $\mathbf{w}$, tj. ako model defniramo kao homogenu funkciju umjesto kao afinu funkciju. No ova razlika je nebitna; nastavljamo s izdvojenom težinom w_0 jer će nam tako biti lakše.

8 Funkcija predznaka ovdje je (što je uobičajeno u strojnom učenju) definirana kao:

$$\text{sgn} : \mathbb{R} \to \{-1, +1\}$$
$$\text{sgn}(x) = \begin{cases} +1 & x \geqslant 0 \\ -1 & x < 0 \end{cases}$$

a ne sa trećom vrijednošću, sgn$(0) = 0$, kako je ta funkcija uobičajeno definirana u matematici.

9 Već smo bili spomenuli, kada smo pričali o linearnim diskriminativnim modelima, da množenje težina $(w_0, \mathbf{w})$ s faktorom $\alpha > 0$ nema utjecaja na položaj i orijentaciju hiperravnine, niti na udaljenost primjera od hiperravnine. Udaljenosti su nepromijenjene budući da:

$$d = \frac{h(\mathbf{x}; \alpha\mathbf{w})}{\|\alpha\mathbf{w}\|} = \frac{\alpha\mathbf{w}^{\mathrm{T}}\mathbf{x} + \alpha w_0}{\|\alpha\mathbf{w}\|} = \frac{\alpha(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)}{\alpha\|\mathbf{w}\|} = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0}{\|\mathbf{w}\|} = \frac{h(\mathbf{x}; \mathbf{w})}{\|\mathbf{w}\|}$$

10 Također primijetite da smo na ovaj način zapravo osigurali korespondenciju 1:1 između parametara hiperravnine $(w_0, \mathbf{w})$ i hipoteze h. Općenito vrijedi da jednu te istu funkciju h možemo definirati s beskonačno mnogo različitih težina $(w_0, \mathbf{w})$, koje se razlikuju samo za konstantan faktor. Uvedenim ograničenjem, međutim, eliminirali smo sve vektore težina osim jednog (onog koji zadovoljava gornje ograničenje).

11 S obzirom da smo objasnili svaki korak u ovoj transformaciji, trebalo bi biti jasno da rješavamo isti optimizacijski problem od kojega smo krenuli, a to je maksimizacija margine uz pretpostavku linearno odvojivih klasa. Međutim, formulacija do koje smo došli opire se intuiciji. Naime, kako to da minimizacija druge norme vektora težina $\mathbf{w}$ (sans w_0), uz navedena ograničenja, daje rješenje maksimalne margine? Pokušajmo to objasniti. Prvo opažanje je da, što je norma težina $\mathbf{w}$ *manja* (tj. što je vektor normale *kraći*), to su primjeri za koje $yh(\mathbf{x}) = 1$ *dalje* od hiperravnine. Zašto je to tako? Zato što je izlaz hipoteze h jednak $\mathbf{w}^{\mathrm{T}}\mathbf{x}$, pa će za neki fiksni $\mathbf{x}$ izlaz biti to manji što je $\mathbf{w}$ manji, a to znači da se primjer za koji $yh(\mathbf{x}) = 1$ onda nalazi dalje od hiperravnine. To zapravo znači da minimizacijom od $\mathbf{w}$ mi efektivo širimo marginu. Naravno, tu su i ograničenja, i ona zahtijevaju da za sve primjere vrijedi $yh(\mathbf{x}) \geqslant 1$, što znači da se primjeri ne mogu nalaziti "unutar margine". Imamo, dakle, dva suprotstavljena zahtjeva: s jedne strane, minimiziramo $\|\mathbf{w}\|$, čime širimo marginu,

s druge strane, imamo ograničenja koja sprječavaju da margina bude tako široka da u nju uđu neki primjeri. Posljedica ovoga je da dobivamo što širu marginu, ali točno takvu da su primjeri koji su najbliži margini oni za koje $yh(\mathbf{x}) = 1$, i svi primjeri su ispravno klasificirani. Ova opozicija između što šire margine i poštovanja ograničenja analogna je argmax-min kriteriju od kojega smo početno krenuli.

12 Nemojte da vas zbuni naziv "programiranje". Ne misli se o programiranju u smislu kodiranja, u C-u ili nedajbože Javi. Radi se o **matematičkom programiranju**, što je stari naziv za **(matematičku) optimizaciju**.

13 Formalno, **konveksni optimizacijski problem** (engl. *convex optimization problem*) definiran je kao:

$$\begin{aligned} \text{minimizirati} \quad & f(\mathbf{x}) \\ \text{uz ograničenja} \quad & g_i(\mathbf{x}) \leqslant 0, \quad i = 1, \ldots, m \\ & \mathbf{a}_i^{\mathrm{T}}\mathbf{x} - b_i = 0, \quad i = 1, \ldots, p \end{aligned}$$

gdje su ciljna funkcija f i ograničenja nejednakosti g_i **konveksne funkcije**, dok su ograničenja jednakosti h_i **afine funkcije**. U ovom slučaju, ograničenje na minimizaciju od $f(\mathbf{x})$ doista predstavlja ograničenje, jer maksimizacija od $-f(\mathbf{x})$ više nije konveksan problem budući da funkcija $-f(\mathbf{x})$ nije konkveksna nego konkavna. Međutim, u praksi nas maksimizacija konveksnih funkcija i minimizacija konkavnih funkcija ne zanimaju. Zanimaju nas minimizacija konveksnih funkcija i maksimizacija konkavnih funkcija, a to je obuhvaćeno gornjom definicijom konveksnoga optimizacijskog problema. Detaljnije u (Boyd et al., 2004) (poglavlje 4).

14 Formalno, **kvadratni program** (engl. *quadatic program, QP*) definiran je kao:

$$\begin{aligned} \text{minimizirati} \quad & \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{x} + \mathbf{q}^{\mathrm{T}}\mathbf{x} + r \\ \text{uz ograničenja} \quad & \mathbf{G}\mathbf{x} \preceq \mathbf{h} \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned}$$

gdje je $\mathbf{P}$ simetrična i realna matrica dimenzija $n \times n$, $\mathbf{q}$ je n-dimenzijski vektor realnih brojeva, $\mathbf{G}$ je realna matrica dimenzija $n \times m$, a $\mathbf{A}$ je realna matrica dimenzija $p \times n$. Notacija $\mathbf{G}\mathbf{x} \preceq \mathbf{h}$ označava da su sve komponente vektora $\mathbf{G}\mathbf{x}$ manje od ili jednake njima odgovarajućim komponentama vektora $\mathbf{h}$. Primijetite da to zapravo znači da su ograničenja nejednakosti (isto kao i ograničenja jednakosti) linearna (odnosno afine funkcije). Detaljnije o kvadratnom programiranju možete pročitati u (Boyd et al., 2004) (poglavlje 4).
Lako se pogubiti među svim ovim varijantama optimizacijskih problema; u snalaženju može pomoći ova taksonomija optimizacijskih problema: https://neos-guide.org/content/optimization-taxonomy. U toj taksonomiji, kvadratno programiranje nalazimo kao deterministički, kontinuirani, ograničeni optimizacijski problem s linearnim ograničenjima. Druga vrsta konveksnog optimizacijskog problema, koji je jednostavniji od kvadratnog programa, a koji se u računarstvu također često koristi, je **linearni program**, kod kojega je ciljna funkcija linearna (ograničenja su istog oblika kao i za kvadratni program).

15 Metodu **Lagrangeovih multiplikatora** osmislio je talijanski matematičar i astronom Joseph-Louis Lagrange u 18. stoljeću. Lagrange je jedan od najvećih matematičara, poznat po nizu važnih doprinosa matematici (u diferencijalnom računu, varijacijskom računu i teoriji brojeva), fizici (teorijska mehanika) i astronomiji (problem triju tijela). Lijep opis Lagrangeova života i stvaralaštva možete pročitati ovdje: https://www.famousscientists.org/joseph-louis-lagrange/. Langrangea smo već bili spomenuli u kontekstu regresije i Lagrangeovog interpolacijskog teorema.

16 **Karush-Kuhn-Tuckerovi (KKT) uvjeti** (engl. *Karush-Kuhn-Tucker (KKT) conditions*) su nužni (ali ne i dovoljni) uvjeti za optimalnost rješenja nelinearnog optimizacijskog problema, pod uvjetom da optimizacijski problem zadovoljava tzv. **uvjeti regularnosti** (engl. *regularity conditions*) (koji ovise o

konkretnom optimizacijskom problemu). Uvjeti KKT su:

$$\begin{aligned} g_i(\mathbf{x}) &\leqslant 0, \quad i = 1, \ldots, m \\ h_i(\mathbf{x}) &= 0, \quad i = 1, \ldots, p \\ \alpha_i &\geqslant 0, \quad i = 1, \ldots, m \\ \alpha_i g_i(\mathbf{x}) &= 0, \quad i = 1, \ldots, m \end{aligned}$$

Prva dva uvjeta su uvjeti **primarne ostvarivosti** (engl. *primal feasiblity*), treći uvjet je uvjet **dualne ostvarivosti** (engl. *dual feasibility*), a četvrti uvjet je već spomenuti uvjet **komplementarne labavosti** (engl. *complementarity slackness*).
Uvjeti KKT su nužni i *dovoljni* uvjeti za optimalnost rješenja ako je optimizacijski problem konveksan (tj. ciljna funkcija f i funkcije ograničenja nejednakosti g_i su konveksne, a funkcije ograničenja jednakosti h_i su afine) te su funkcije g_i derivabilne. KKT uvjeti zapravo proširuju metodu Lagrangeovih multiplikatora, koja dopušta samo ograničenja jednakosti. Budući da mi ovdje koristimo i uvjete nejednakosti, točnije bi bilo reći da koristimo optimizacijsku metodu KKT, a ne metodu Lagrangeovih multiplikatora.
Uvjete KKT predložili su matematičari Harold Kuhn i Albert Tucker 1951. godine, ali ih je William Karush objavio još kao student desetak godina ranije u svojem diplomskom radu (za što je nagrađen s prvim "K" u kratici KKT).

17 Kod konveksnih optimizacijskih problema, uvjet regularnosti (koji je preduvjet za uvjete KKT) jest da postoji točka $\mathbf{x}$ takva da $h(\mathbf{x}) = 0$ i $g_i(\mathbf{x}) < 0$, gdje je h ograničenje jednakosti, a g_i ograničenje nejednakosti. To je tzv. **Slaterov uvjet regularnosti**. Slaterovi uvjeti ujedno su dovoljni uvjeti za jaku dualnost konveksnog optimizacijskog problema.

18 Za nestrpljive: rješavanje problema maksimalne margine SVM-a u dualnoj formulaciji ima tri važne prednosti nad rješavanjem tog problema u primarnoj formulaciji. Prvo, u dualnoj formulaciji može se primijeniti algoritam **slijedne minimalne optimizacije** (engl. *sequential minimal optimization, SMO*), koji je učinkovit algoritam specijaliziran za kvadratno programiranje baš za SVM. (Mi, nažalost, nećemo imati vremena pričati o tom algoritmu.) Drugo, u dualnoj formulaciji model ćemo moći definirati preko tzv. **potpornih vektora**, što ima svoje prednosti u pogledu računalne učinkovitosti. Treće, u dualnoj formulaciji moći ćemo koristiti tzv. **jezgreni trik**, i time zaobići potrebu da ručno i eksplicitno definiramo funkciju preslikavanja. Mogućnost korištenja jezgrenog trika zapravo je razlog zašto je SVM tako moćan i popularan algoritam.

19 Algoritam **slijedne minimalne optimizacije** (engl. *sequential minimal optimization, SMO*) osmislio je američki računalni znanstvenik John Platt 1998. godine (Platt, 1998). Platt, koji trenutačno radi u Googleu, osim po algoritmu SMO i tome da je otkrio dva asteroida promjera oko 25 km, poznat je i po metodi **Plattovog skaliranja**, kojom omogućava da se izlaz SVM-a vrlo jednostavno transformira u probabilistički izlaz u intervalu $(0, 1)$. Više o tome idući put.

20 Težinu w_0 možemo izračunati na temelju činjenice da za potporne vektore vrijedi $y^{(i)}h(\mathbf{x}^{(i)}) = 1$. Neka je S skup indekasa potpornih vektora $\mathbf{x}^{(i)}$. Ako u $y^{(i)}h(\mathbf{x}^{(i)}) = 1$ uvrstimo

$$h(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y^{(i)} (\mathbf{x})^{\mathrm{T}} \mathbf{x}^{(i)} + w_0$$

onda dobivamo:

$$y^{(i)}h(\mathbf{x}^{(i)}) = y^{(i)} \Big(\sum_{j \in S} \alpha_j y^{(j)} (\mathbf{x}^{(i)})^{\mathrm{T}} \mathbf{x}^{(j)} + w_0 \Big) = 1$$

Iz toga dobivamo:

$$w_0 = y^{(i)} - \sum_{j \in S} \alpha_j y^{(j)} (\mathbf{x}^{(i)})^{\mathrm{T}} \mathbf{x}^{(j)}$$

pri čemu smo iskoristili $1/y^{(i)} = y^{(i)}$ jer $y^{(i)} \in \{-1, +1\}$. Ovu jednadžbu možemo izračunati na temelju jednog, proizvoljno odabranog označenog primjera $(\mathbf{x}^{(i)}, y^{(i)})$. Međutim, zbog numeričkih odstupanja nećemo za svaki odabir dobiti isto rješenje, pa je bolje izračunati prosjek nad svim potpornim vektorima:

$$w_0 = \frac{1}{|S|} \sum_{i \in S} \Big(y^{(i)} - \sum_{j \in S} \alpha_j y^{(j)} (\mathbf{x}^{(i)})^{\mathrm{T}} \mathbf{x}^{(j)} \Big)$$

Literatura

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

J. Mount. How sure are you that large margin implies low vc dimension? *Win-Vector Blog*, 2015.

J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.