Eötvös Loránd Tudományegyetem

Természettudományi Kar

Nagy Levente

Szupport vektor gép konstrukciók összehasonlítása

Szakdolgozat

Matematika BSc

Témavezető:
Csáji Balázs Csanád
Valószínűségszámításelméleti és Statisztika Tanszék

Budapest, 2025

Köszönetnyilvánítás

Szeretném kifejezni őszinte köszönetemet és hálámat témavezetőm, Csáji Balázs Csanád felé az egész félév során tartó közös munkáért. Nagy segítséget kaptam egészen a közös témameghatározástól kezdve, a folyamatos konzultációk során Tanár Úr szaktudásának hála még mélyebb betekintést nyerhettem a témába. Az értékes tanácsoknak, iránymutatásoknak köszönhetően a kisebb akadályok vagy olykor zsákutcák is sikeresen elhárultak. Mindenkinek ilyen konzulenst szeretnék kívánni.

Tartalomjegyzék

Bevezetés

A gépi tanulás a mesterséges intelligencia egyik legfontosabb és legdinamikusabban fejlődő ága. Olyan algoritmusok és modellek fejlesztésével foglalkozik, amelyek képesek az adatokból tanulni, majd ezek után következtetések és döntések meghozatalára. A gépi tanulásnak 3 fő típusa van: felügyelt tanulás, felügyeletlen tanulás és a megerősítéses tanulás. Én ebben a dolgozatban a felügyelt tanulás témakörével foglalkozok, ahol $\mathcal{X}$ halmazbeli elemekhez próbálunk egy-egy $\mathcal{Y}$ halmazbeli értéket rendelni. Rendelkezésünkre áll egy n elemű minta (ez lesz a tanító adatunk), $\{(x_j, y_j)\}_{j=1}^{n} \in \mathcal{X} \times \mathcal{Y}$ ennek segítségével szeretnénk megmondani, hogy még nem látott x értékekhez milyen y tartozhat. Ha $\mathcal{Y}$ elemszáma véges, akkor osztályozási (klasszifikációs) feladatról beszélünk, más esetben regressziósról (ekkor valamilyen folytonos értéket próbálunk becsülni). Egy tipikus osztályozási feladat, mikor az emailek egy nagy halmazából próbáljuk kiszűrni a spam emaileket, egy jellegzetes regressziós probléma az ingatlanbecslés, egy ingatlan értékét próbáljuk becsülni a tulajdonságai (alapterület, lokáció, szobaszám stb.) alapján. A tanulás során megengedett függvények halmaza, a modellosztály $\mathcal{M}$, az $f : \mathcal{X} \to \mathcal{Y}$-az ilyen f függvének összessége. Ebből a halmazból szeretnénk kiválasztani azt a függvényt ami a legjobb becslést adja y-ra tetszőleges $x \in \mathcal{X}$ esetén.

Az adatok osztályozása, mint például képosztályozás vagy a kézzel írt karakterek felismerése és kiértékelése ma különösen fontos (példának okáért a ChatGPT-nek is ez volt az egyik legfontosabb behozott funkciója). Ha az adatpontjaink d dimenziós vektorunk és azt vizsgáljuk, létezik-e $d-1$ dimenziós hipersík, amely elválasztja a pontokat. Ez a lineáris osztályozás. Számos hipersík alkalmas lehet a pontok szétválasztására, de ésszerű választás az a hipersík, amely a két osztály közti legnagyobb távolságot biztosítja. Ez a margó. Ennek felel meg az a választás, mikor mindkét oldalon a legközelebbi adatpont minél távolabb van. Az így definiált lineáris osztályozót nevezzük szupport vektor gépnek. A szupport vektor gépek először bináris osztályozási feladatra lettek kitalálva, majd kiterjesztették több osztályozás és regressziós feladatokra is. Ebben a szakdolgozatban főleg

a bináris klasszifikációs részével foglalkozunk.
A szupport vektor gépek elméleti alapjait Vapnik (1982, 1995) és Chervonenkis (1974) rakták le. Az utóbbi két évtizedben az egyik legkutatottabb felügyelt tanulásos modellé váltak.
A szupport vektor gépek esetében a modell paramétereinek meghatározása konvex optimalizálási problémává redukálódik. Ennek számos előnye van, többek között az, hogy csak globális optimumunk van, így ha találunk megoldást, az biztosan a legjobb lesz, valamint használhatjuk a konvex optimalizálás már kifejlesztett hatékony megoldó algoritmusait.

A lineáris modellek előnye, hogy gyorsan számíthatóak, azonban velük csak lineáris összefüggéseket tudunk felismerni. Nemlineáris problémákat is meg tudunk velük oldani, ha az adatpontokat egy Φ leképezés segítségével felvetítjük képtérbe, amit tulajdonságtének fogunk nevezni, s itt alkalmazzuk a modellt. Magasabb dimenzióban nagyon költséges lehet kiszámolni minden pontra expliciten a projekciót. Erre nyújt megoldást a reprezentációs tétel, miszerint az f döntési függvény felírható a tanuló adatpontok lineáris kombinációjaként, $f(x) = \sum_{i=1}^{n} \alpha_i \langle \Phi(x), \Phi(x_i) \rangle$. Az f meghatározásához szükségünk van α-ra. Sokat tudunk egyszerűsíteni a számításokon azzal, ha a skaláris szorzatok kiszámolásán gyorsítunk azzal, hogy a skaláris szorzatokat egy problémához illeszkedő kernel függvénnyel definiáljuk, ami ezt az explicit képletet adja $k(x_i, x) = \langle \Phi(x_i), \Phi(x) \rangle$.

Egy gyakorlati probléma sok kernel függvényeken alapuló módszerrel, hogy a kernel függvényeket minden lehetséges x_i, x_j tanulópont párra ki kell számítani. Ez új adatpontra történő predikció esetén túlzott számítási időt igényel. Ezzel szemben a szupport vektor gépek előnye, hogy ritkásabb megoldásokkal rendelkeznek, az új bemenet osztályozása csak a minta adathalmaz egy részhalmazán kiszámított kernel függvényektől függ.

1. fejezet

Statisztikai tanuláselmélet alapok

A klasszifikáció a statisztikai tanuláselmélet egyik alapvető problémája, amely számos gyakorlati feladatban megjelenik. A bináris klasszifikáció során adott n elemű, független, azonos eloszlású mintánk $\{(x_j, y_j)\}_{j=1}^n$, az (X, Y) véletlen vektor ismeretlen $\mathbb{P}$ eloszlásból származtatva, ahol x_i az i-edik bemenet és y_i a hozzá tartozó címke, valamint $x_i \in \mathcal{X} \subseteq \mathbb{R}^d$ és $y_i \in \mathcal{Y} = \{-1, +1\}$. Osztályozónak nevezünk minden $f : \mathcal{X} \to \{-1, +1\}$ alakú (mérhető) leképezést.

Ahhoz, hogy tudjuk mérni a becslések jóságát, szükségünk van egy függvényre, amely nyomon követi adott $x \in \mathcal{X}$ esetén az $f \in \mathcal{M}$ mennyit téved. Az $\mathcal{X}$ és $\mathcal{Y}$ tetszőleges mérhető halmazok.

1.0.1. Definíció. *[1, 3.1 Definition] Adott egy $x \in \mathcal{X}$, $y \in \mathcal{Y}$ minta, $f(x) \in \mathcal{Y}$ becslés y-ra, amelyet az x-ből készítettünk. Jelölje $(x, y, f(x)) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ pedig az ezek által alkotott triót. Jelölje az $(x, y, f(x))$ hármast egy minta x, egy megfigyelés y, és a $f(x)$ predikció alkotta hármas. Ekkor az $\ell : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ leképezést veszteségfüggvénynek nevezzük, ha teljesül, hogy $\ell(x, y, y) = 0$ minden $x \in \mathcal{X}$ és $y \in \mathcal{Y}$ esetén.*

Megköveteljük, hogy ℓ nemnegatív függvény legyen. Ekkor biztosítható az is, hogy a $f(x) = y$ pontos egyezés esetén ne szülessen veszteség. Fontos kitétel azonban, hogy nincs univerzálisan legjobb veszteségfüggvény, amely minden feladatban elsőbbséget élvez, veszteségfüggvény választása mindig az adott feladat körülményeitől függ.

Miután meghatároztuk, hogyan büntetjük a hibákat, összegeznünk kell őket. Mostantól feltételezzük, hogy létezik egy $\mathcal{P}(x, y)$ valószínűségi eloszlás $\mathcal{X} \times \mathcal{Y}$-on, amely meghatározza az adatgenerálást. Ha ismerjük a teszt adatokat is, akkor a cél az, hogy specifikusan ezeken a lehető legjobb eredményt érjük el. Ha nem ismerjük, akkor a célünk az,

hogy az összes lehetséges teszt mintán általánosságban akarjuk minimalizálni a hibákat.

1.0.2. Definíció. *[1, 3.2 Definition] Tegyük fel, hogy a tanító adatokon $(x_1, \ldots, x_m)$ és célértékeken $(y_1, \ldots, y_m)$ kívül a teszt adatokat is ismerjük: $(x'_1, \ldots, x'_{m'})$. Az ezeken várható minimális hiba*

$$R_{test}[f] := \frac{1}{m'} \sum_{i=1}^{m'} \left(\int_{\mathcal{Y}} \ell(x'_i, y, f(x'_i)) \, dP(y \mid x'_i) \right)$$

Ez a probléma azonban elméletileg és számítás szempontjából is nehéz. Ehelyett jellemzőbb feladat az, ha a tesztmintákat nem ismerjük.Ha nem ismerjük (vagy figyelmen kívül hagyjuk) a tesztmintákat, akkor a cél az, hogy az összes lehetséges minta átlagában minimalizáljuk a veszteséget.

1.0.3. Definíció. *[1,3.3 Definition] Ha $\mathcal{X} \times \mathcal{Y}$ egy mérhető tér, amely felett definiálva van $\mathcal{P}$ eloszlás és $\ell(x, y, f(x)$ mérhető veszteségfüggvény, akkor a minimalizálandó érték a kockázat, a veszteségfüggvény várható értéke.*

$$R[f] := \mathbb{E}[R_{test}[f]] = \mathbb{E}[\ell(x, y, f(x))] = \int \ell(x, y, f(x)) \, dP(x, y)$$

Bináris osztályozásnál egy tipikus választás a 0-1 veszteségfüggvény, ami $\ell(x, y, f(x)) = \mathbb{I}(f(x) \neq y)$ ahol $\mathbb{I}$ az indikátorfüggvény. Ebben az esetben az a priori kockázat a félreosztályozás valószínűsége: $R(f) = \mathbb{P}(f(X) \neq Y)$.

A feladat nehézségét az adja, hogy egy olyan mennyiséget próbálunk minimalizálni, mivel nem ismerjük $\mathbb{P}$-t, az integrált sem tudjuk közvetlenül kiszámítani. Amit ismerünk, az $\{(x_j, y_j)\}_{j=1}^n$ tanulási minta amelyet $\mathbb{P}$-ből alkottunk. Így az eloszlást közelíthetjük a tanító halmaz segítségével, úgy, hogy a minta szerinti diszkrét eloszláson, azaz $P_{\text{emp}}(x, y) = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}(x) \, \delta_{y_i}(y)$ nézzük a veszteségfüggvény várható értékét, ahol $\delta_a(v) = 1$ ha $a = v$, és 0 egyéb esetben.

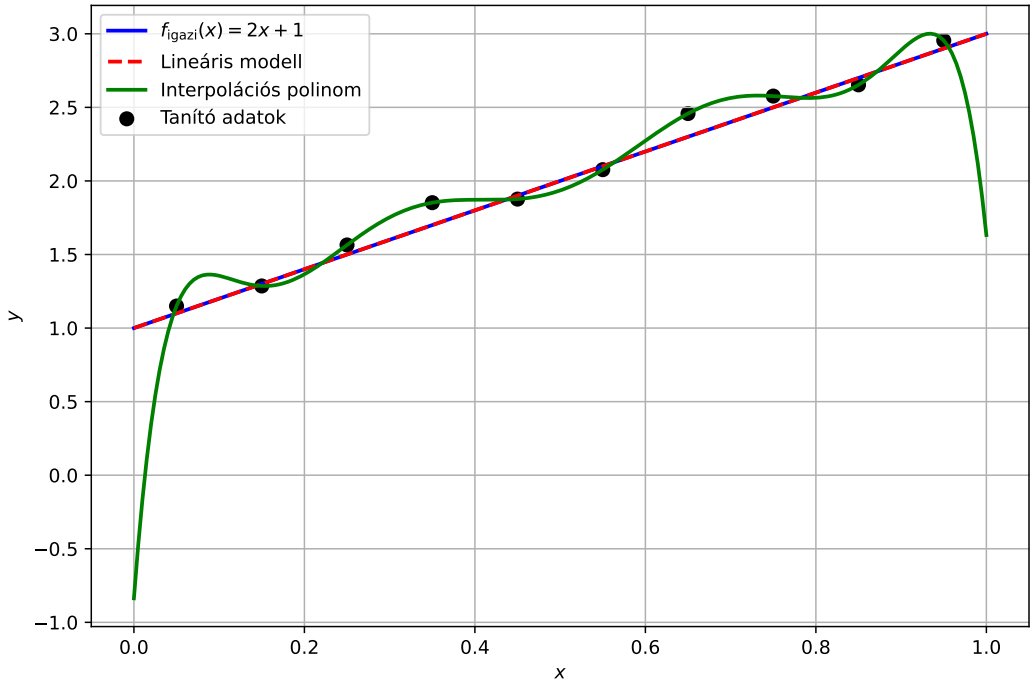
1.0.4. Definíció. *[1,3.4 Definition] Az ez alapján definiált empirikus kockázat egy már könnyen kalkulálható érték:*

$$R_{emp}[f] := \int_{\mathcal{X} \times \mathcal{Y}} \ell(f(X), Y) \, \mathrm{d}P_{emp}(x, y) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

Az empirikus kockázattal kapcsolatban azonban ügyelni kell arra, hogy $\mathcal{M}$-et milyennek választjuk. Ha túl nagynak választjuk a megengedhető függvényosztályt, találunk olyan

f-et ami kis empirikus kockázatot ad. Ha minden $\mathcal{X} \to \mathcal{Y}$ függvényt megengedünk, akkor minimális empirikus kockázatot érhetünk el, de ez nem jelenti azt, hogy a minimális kockázathoz. Erre egy jó példa: $f(x) = y_i$, ha $x = x_i$ valamelyik tanítópont, különben $f(x) = 0$. Ez 0 empirikus kockázatot eredményez. Ez azonban nem jelent valódi tanulást, ha kapunk egy új, tanító adatban nem szereplő x-et, akkor szinte biztos, hogy egyik tanító adatbeli x_i-vel se fog megegyezni. Így a függvény csupa 0-s tippelése nem hoz jobb eredményt, mint a puszta véletlen. Egy másik klasszikus példa, mikor az n elemű mintára a síkon interpolációval egy $n-1$-ed fokú polinomot illesztünk.

1.0.5. Példa. *[interpoláció] Legyen a valódi függvényünk $y = 2x + 1$, de a mérések zajosak, így ilyen tanító adatok keletkeztek: $y_i = 2x_i + 1 + \varepsilon_i$, $x \sim \mathcal{U}([0,1])$, $\varepsilon \sim \mathcal{N}(0, 0.1)$. Ezt szemlélteti az 1.1. ábra is.*



1.1. ábra. Lineáris modell vs. Interpolációs polinom

Bár a magas fokú polinom az empirikus kockázat minimalizálása szempontjából a legjobb (a tanító adatokra zérus hibát ér el), ez az eredmény megtévesztő, mert a modell

túltanul és rosszul teljesít a teszt adatokon. Ezzel szemben a lineáris modell nagyobb empirikus hibával, de jobb általánosítást biztosít, mert nem illeszkedik túlságosan a zajhoz. Ezekből azt a tanulságot vonhatjuk le, hogy alacsony empirikus rizikóval nem feltétlen jár együtt az alacsony kockázat is.

Ezt a problémát az induktív torzítással tudjuk orvosolni. Ennek alapvetően 2 módja van. Választhatunk egy szűkebb függvényosztályt $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, amely korlátozza a modell kapacitását. Másik opció az, hogy alapvetően szélesebb függvényosztályt engedünk meg, de büntetőtaggal büntetjük a komplexitást. Gyakorlatban a megközelítés gyakran együttesen alkalmazandó: először egy korlátozott kapacitású függvényosztályt választunk (pl. folytonos, lineáris vagy sima függvényeket engedünk meg), majd egy regularizációs taggal – mint például az L^2 norma – finomhangoljuk a preferenciát az egyszerűbb modellek irányába.
A regularizált tanulási feladat célja a regularizált kockázat minimalizálása.

1.0.6. Definíció. *[1, 4.1] A regularizált kockázat az empirikus kockázat és egy regularizációs funkcionál összege:*

$$R_{reg}[f] := R_{emp}[f] + \Omega[f],$$

ahol $\Omega : \mathcal{F}(X, Y) \to [0, \infty)$ *regularizációs funkcionál segítségével tudjuk büntetni* $f \in \mathcal{F}(X, Y)$ *komplexitását. Ennek egy alternatív formája:*

$$R_{reg}[f] := R_{emp}[f] + \lambda\Omega[f],$$

ahol $\lambda > 0$ *regularizációs paraméter, ami megadja az empirikus kockázat és a modell egyszerűsége közti egyensúlyt.*

Ha $R_{\text{emp}}[f]$, akkor célszerű $\Omega[f]$-t is konvexnek választani, mert így egy konvex optimalizálási problémát kapunk. Most rátérünk az induktív torzítás másik módjára. Fontos, hogy milyen $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ függvényosztály felett minimalizáljuk az empirikus kockázatot. Vizsgáljuk a függvényosztály kapacitását (komplexitását). Ha túl nagy a kapacitás, akkor a modell képes lehet túltanulni, túlságosan a tanító adatokat tanulja. Ha viszont túl kicsi a kapacitás, a modell nem lesz képes megtanulni a valódi összefüggéseket, vagyis alultanul. A statisztikai tanuláselméletben több koncepció is létezik a kapacitásfogalomra (például: entrópia, lefedési számok, VC-dimenzió). Én a VC-dimenziót muatatom be részletesebbről.

1.0.7. Definíció. *[1, 5.5.3] Legyen* $Z_n := ((x_1, y_1), \ldots, (x_n, y_n))$ *egy* n *elemű minta* $\mathcal{X} \times \mathcal{Y}$*-ből,* $\mathcal{Y} = \{+1, -1\}$*,* $\mathcal{F} \subseteq \mathcal{F}(\mathcal{X}, \mathcal{Y})$ *pedig egy függvényosztály. Jelöljük* $\mathcal{N}(\mathcal{F}, Z_n)$*-nel*

azon $\mathcal{F}$ osztálybeli függvények számát, amelyeket meg tudunk különböztetni az $x_1, x_2, \ldots, x_n$ pontokon felvett értékeik alapján. Ennek maximumát az összes lehetséges n elemű mintán jelöljük $\mathcal{N}(\mathcal{F}, n)$-nel. Ezt törő együtthatónak nevezzük.

Ez megadja egy adott $x_1, x_2, \ldots, x_n$ ponthalmaznak hány különféle $y_1, y_2, \ldots y_n$ címkézése van, melyre tudunk találni egy $\mathcal{F}$-beli függvényt, ami ezeket helyesen osztályozza. Bináris klasszifikációnál mind az n pontot csak 2-féle osztályba tudjuk sorolni, szóval $\mathcal{N}(\mathcal{F}, n) \leq 2^n$, amikor $\mathcal{N}(\mathcal{F}, n) = 2^n$ felvétetik, azt mondjuk $\mathcal{F}$ széttör n pontot.
Egy halmazosztály $\mathcal{C}$ széttör egy A halmazt, ha $\forall S \subseteq A : \exists C \in \mathcal{C}$ úgy, hogy $S = A \cap C$. Egy osztályozók osztálya $\mathcal{F}$ széttör egy $A \subseteq X$ halmazt, ha az $\mathcal{F}$ által generált halmazosztály, legyen ez $\mathcal{C}_{\mathcal{F}} := \{C \subseteq X : \exists f \in \mathcal{F},\ C = f^{-1}(\{1\})\}$ széttöri A-t. A VC-dimenzió az osztályozóképesség mértéke.

1.0.8. Definíció. *[1, 5.5.6] Egy $\mathcal{F}$ függvényosztály VC-dimenziója az a* $\mathrm{VC}_{\dim}(\mathcal{F}) = \max\{h \in \mathbb{N} : \exists A \subseteq X,\ |A| = h$ *és A-t széttöri $\mathcal{F}$}*. Vagyis $\mathcal{N}(\mathcal{F}, h) = 2^h$, de $\mathcal{N}(\mathcal{F}, h+1) < 2^{h+1}$.*

VC dimenzió segítségével tehetünk egy valószínűségi alapú állítást a kockázat becsléséről.

1.0.9. Állítás. *[1, 1.19 és 1.20] Tegyük fel, hogy $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, $\mathcal{F}$ függvényosztályban dolgozunk, ahol $h < n$ VC dimenzió adott. Ekkor legalább $1 - \delta$ valószínűséggel igaz:*

$$R[f] \leq R_{emp}[f] + \epsilon(h, n, \delta),$$

ahol

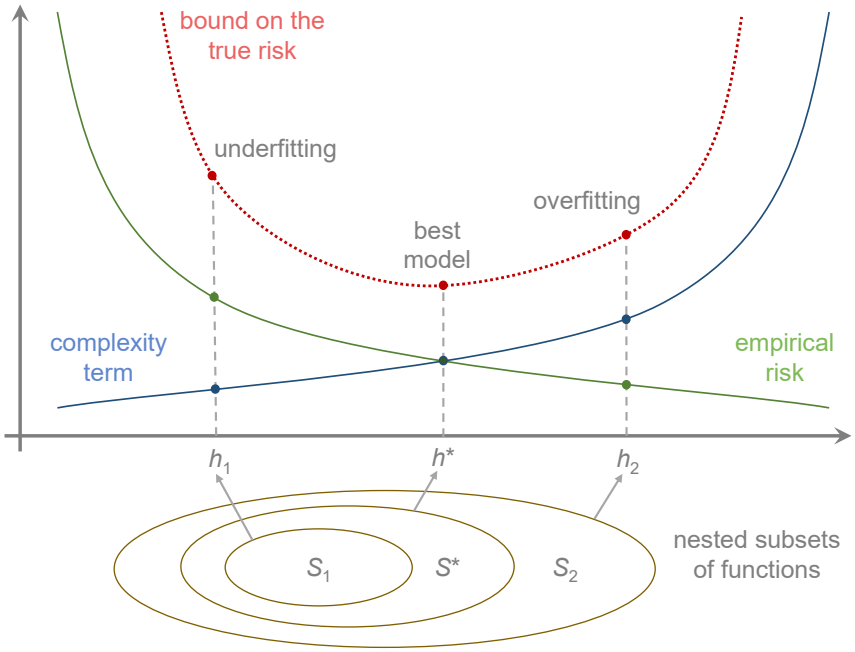
$$\epsilon(h, n, \delta) = \sqrt{\frac{1}{n}\left(h\left(\ln\left(\frac{2n}{h}\right) + 1\right) + \ln\left(\frac{4}{\delta}\right)\right)}$$

komplexitási tag.

A komplexitási tag a VC-dimenzió (h), a becslés igaz valószínűségének ($1 - \delta$) növelésével növekszik, míg a mintaszám növelésével (n) csökken. Intuíció alapján is ez adódik. Az egyenlőtlenségből adódik, hogy nemcsak az empirikus kockázatot szeretnénk minimalizálni, hanem az empirikus kockázatot a komplexitási taggal együtt.

A strukturális kockázatminimalizálás egy módja a tanulásnak, amely által megkaphatjuk a legjobban általánosító modellt. Vegyük egymásba ágyazott függvényosztály

egy láncát $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots \subset \mathcal{F}_n$. Növekvő (de véges) VC dimenzióval, vagyis kapacitással $\mathcal{F}_i$-be tartozó függvények egyre bonyolultabbak.



1.2. ábra. Strukturális kockázatminimalizálás [6]

Ahogy 1.2 is mutatja, ha túl egyszerű a modell (kis kapacitás avagy komplexitás), akkor a modell nem tanul jó, nagy tanítási hibát kapunk (underfitting). Ha túl bonyolult a modell (nagy kapacitás), akkor a tanító adatokon kis tanítási hibát kapunk, de a teszt adatokon rossz eredményt érünk el (overfitting). Ez a megközelítés segít megtalálni az arany középutat a komplexitási tag és az empirikus kockázat minimalizálása között, hogy megkapjuk azt a modellt, amelynek a legkisebb felső korlátja van a kockázatra nézve.
Dolgozzunk egy $\mathcal{F}$ skalárszorzatos vektortérben, vegyünk egy olyan függvényosztályt, ahol a döntési függvények a következő alakú hipersíkok: $\langle w, x\rangle + b = 0$, ahol $w \in \mathcal{F}$, $b \in \mathbb{R}$. Az osztályozó függvények $f(x) = \operatorname{sgn}(\langle w, x\rangle + b)$.

1.0.10. Tétel. *[1, 5.5 Theorem] $x_1, x_2, \ldots, x_n \in \mathcal{X}, \mathcal{Y} = \{+1, -1\}$ vegyük azokat a hipersíkokat, amelyek $\langle w, x\rangle = 0$ alakban írhatóak le, a távolságot úgy állítsuk be, hogy minden pont legalább 1 távolságra van a hipersíktól, kanonikus alakot használunk,* $\min_{i=1,\ldots,n} |\langle w, x_i\rangle| = 1$*, vagyis minden tanítóadatra $y_i(\langle w, x_i\rangle) \geq 1$ teljesül. $\mathcal{X}$-n értelmezett osztályozási függvények $f_w = \operatorname{sgn}(\langle w, x\rangle)$, legyen $\|w\| \leq \Lambda$, R az a legkisebb sugarú, origó középpontú kör sugara, amely tartalmazza az $x_1, x_2, \ldots, x_n$ pontokat. Ekkor a VC dimenzió $h \leq R^2\Lambda^2$.*

Bizonyítás. [1, 5.5 Proof] A kanonikus reprezentációból jön, hogy $y_i(\langle w, x_i\rangle) \geq 1$ $i = 0, 1, \ldots, n$-re. Ezt összegezve kapjuk

$$\langle w, \sum_{i=1}^{n} y_i x_i\rangle \geq n.$$

A Cauchy-Schwarz-egyenlőtlenséget használva

$$\langle w, \sum_{i=1}^{n} y_i x_i\rangle \leq \|w\| \cdot \left\|\sum_{i=1}^{n} y_i x_i\right\| \leq \Lambda \cdot \left\|\sum_{i=1}^{n} y_i x_i\right\|.$$

A második egyenlőtlenségnél $\|w\| \leq \Lambda$ feltételt használtuk ki. A $\langle w, \sum_{i=1}^{n} y_i x_i\rangle$-re kapott alsó és felső korlátot használva:

$$n \leq \Lambda \left\|\sum_{i=1}^{n} y_i x_i\right\| \Longrightarrow \frac{n}{\Lambda} \leq \left\|\sum_{i=1}^{n} y_i x_i\right\|$$

$y_i \in \{+1, -1\}$, y_i-k egymástól függetlenek, s egyenletes eloszlású változók, így $E[y_i] = 0$. Ha $i \neq j$, akkor $\mathbb{E}[y_i y_j] = \mathbb{E}[y_i] \cdot \mathbb{E}[y_j] = 0 \cdot 0 = 0$. Ha $i = j$, akkor $\mathbb{E}[y_i^2] = \mathbb{E}[1] = 1$, mert $y_i^2 = 1$ mindig fennáll.
Számoljuk ki $\|\sum_{i=1}^{n} y_i x_i\|^2$ várható értékét. Számolás közben használhatjuk a várható érték linearitását

$$\mathbb{E}\left[\left\|\sum_{i=1}^{n} y_i x_i\right\|^2\right] = \sum_{i=1}^{n} \mathbb{E}\langle y_i x_i, \sum_{i=j}^{n} x_j y_j\rangle = \sum_{i=1}^{n} \mathbb{E}\langle y_i x_i, \sum_{i\neq j} y_j x_j\rangle + \sum_{i=1}^{n} \mathbb{E}\langle y_i x_i, y_i x_i\rangle =$$

$$\sum_{i=1}^{n}\left(\sum_{i\neq j} \mathbb{E}\ \langle y_i x_i, y_j x_j\rangle\right) + \sum_{i=1}^{n} \mathbb{E}\|y_i x_i\|^2 = \sum_{i=1}^{n} \mathbb{E}\|y_i x_i\|^2$$

Mivel $\|y_i x_i\| = \|x_i\|$, s $\|x_i\| \leq R$ (feltétel), így $\mathbb{E}\left\|\sum_{i=1}^{n} y_i x_i\right\|^2 \leq nR^2$. A várható érték a címkék véletlenszerű címkézésére igaz, így lesz legalább 1 olyan címkézés, amelyre igaz $\|\sum_{i=1}^{n} y_i x_i\|^2 \leq nR^2$. Ezt a címkézést használva adódik

$$\frac{n^2}{\Lambda^2} \leq \left\|\sum_{i=1}^{n} y_i x_i\right\|^2 \leq nR^2 \Longrightarrow n \leq \Lambda^2 R^2.$$

□

Ha n pontot osztályozni tudunk,vagyis szétválasztjuk, akkor n kielégiti ezt az egyenletet. A h VC-dimenzió szintén teljesíti az egyenletet, mivel ez a maximális számú pont, amit szét lehet választani. A tételből jön, hogy a VC-dimenziót a tér dimenziójától

függetlenül kontrollálhatjuk w szabályozásával.

Fogalmazzuk át a tételt egy gyakorlati szempontból jobban értelmezhető formájába. Vegyük ki a $\|w\| \leq \Lambda$ feltételt. A hipersíkjaink $\langle w, x\rangle + b = 0$ alakban írhatóak, a pontokat δ-margónal szeretnénk szeparálni, vagyis $y_i(\langle w, x_i\rangle + b) \geq \delta \ i = 1, 2, \ldots, n$-re. Legyen $\|w\| = 1$ most a feltételünk.A kanonikus alakhoz $\min_{i=1,\ldots,n} y_i\langle w, x_i\rangle = 1$ kellene. Ehhez le kell osztani δ-margónal: $y_i(\langle \frac{w}{\delta}, x_i\rangle + \frac{b}{\delta}) \geq 1$, ez pedig pont a kanonikus forma, $w' = \frac{w}{\delta}$, $\|w'\| = \frac{1}{\delta}$. A bebizonyított tételt alkalmazva $h \leq R^2\|w'\|^2 = \frac{R^2}{\delta^2}$.

1.0.11. Állítás. *[1,5.5.6] Ha $\mathcal{X} = \mathbb{R}^d$ és a függvényosztály a hipersík osztályozókból áll*

$$\mathcal{F} = \left\{\operatorname{sgn}(\langle w, x\rangle + b) : w \in \mathbb{R}^d,\, b \in \mathbb{R}\right\},$$

Ekkor $\mathcal{F}$ VC-dimenziója $d + 1$.

1.0.12. Tétel. *[6] Tegyük fel, hogy a minta adatok lineárisan elválaszthatóak, azaz $y_i(\langle wx_i\rangle + b) \geq \delta$ minden $i = 1, \ldots, n$ esetén, ahol $\delta > 0$ és teljesül, hogy $\|w\| = 1$. Tegyük fel, hogy $x \in X \subseteq \mathbb{R}^d$ egy r sugarú gömbben helyezkedik el. Ekkor a δ-margó szeparáló hipersíkok halmazának VC dimenziója korlátos ekképpen:*

$$h \leq \min\left(\frac{r^2}{\delta^2},\, d\right) + 1.$$

Ekkor a strukturális kockázatminimalizálás alapján, hogy minimalizáljuk a komplexitási tagot, maximalizálnunk kell a margót, s így jutunk el a margó alapú lineáris osztályozók konstrukciójához, a szupport vektor gépekhez. A maximális margónal rendelkező optimális hipersík egyébként egyértelműen létezik.

2. fejezet

Konvex optimalizálás

Röviden átveszem a szupport vektor gépek megoldásához szükséges konvex optimalizálási ismereteket. A tanulás során gyakran egy kockázati funkcionált $R_{\text{emp}}[f]$ vagy $R_{\text{reg}}[f]$-t szükséges minimalizálni. Egy tetszőleges függvény minimalizálása egy adott tartományon általában nehéz feladat jó eséllyel sok lokális minimummal. Ezzel szemben egy konvex célfüggvény egy konvex tartományon történő optimalizálása pontosan egy globális minimumot eredményez.

2.0.1. Definíció. *[1,6.1 Definiton] Egy $X \subset \mathcal{X}$ halmaz konvex ($\mathcal{X}$ egy vektortér), ha*

$$\forall x, x' \in X,\ \forall \lambda \in [0,1]:\ \lambda x + (1-\lambda)x' \in X.$$

2.0.2. Definíció. *[1,6.2 Definition] A $f : X \to \mathbb{R}$ függvény konvex, ha*

$$\forall x, x' \in X,\ \forall \lambda \in [0,1] : \lambda x + (1-\lambda)x' \in X \Rightarrow f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda) f(x').$$

f szigorúan konvex, ha az egyenlőtlenség szigorú egyenlőtlenséggel teljesül.

2.0.3. Tétel. *[1, 6.5 Theorem] Ha egy $f : X \to \mathbb{R}$ konvex függvénynek minimuma van egy $X \subset \mathcal{X}$ konvex halmazon, akkor azok az $x \in X$ pontok, melyeken a minimum felvétetik, egy konvex halmazt alkotnak. Ha f szigorúan konvex függvény, akkor ez a halmaz egyetlen elemet tartalmaz.*

A tétel egyik alkalmazása a feltételes konvex optimalizálási probléma.

2.0.4. Következmény. *[1, 6.6 Corollary] Adottak $f, c_1, c_2, \ldots, c_n$ konvex függvények $X \to \mathbb{R}$ hozzárendelés szerint, X egy konvex halmaz. Minimalizáljuk f függvényt X halmazon $c_i x \leq 0$ feltételek mellett*

$$\arg\min_{x \in X} f(x) \quad \textit{úgy, hogy} \quad c_i(x) \leq 0 \ \textit{minden}\ i \in \{1, 2, \ldots, m\}.$$

Továbbá engedelyézett hozzávenni még a feladathoz $h_j x = 0 \quad \forall j \in \{1, 2 \ldots, p\}$ *feltételeket.*

Legyen $X = \mathbb{R}^n$. Továbbiakban ezt a feladatot nevezzük primál feladatnak. Ha nincs semmilyen megszorítás, és ha f differenciálható, akkor könnyű dolgunk van, egyszerűen csak megoldjuk az $\partial_x f(x) = 0$ egyenletet. Ha azonban már vannak $c_i \leq 0$ feltételek, akkor változik a helyzet. Előfordulhat, hogy az a pont, ahol $\partial_x f(x) = 0$, nem felel meg legalább az egyik korlátnak, így nincs benne az érvényes megoldások halmazában. Más megközelítést kell alkalmazni a szigorú korlátozások helyett. Próbáljuk meg súlyokkal büntetni a feltételek rosszul teljesülését. A Lagrange-függvények pont így közelítik meg a problémát.

2.0.5. Definíció. *[3,5.1.1] A primál feladathoz tartozó Lagrange-függvény felírás*

$$L(x, \alpha) = f(x) + \sum_{i=1}^{m} \alpha_i c_i(x)$$

ahol $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ *minden koordinátában nemnegatív. Ha a primál feladatban* $h_j(x) = 0$ *feltételeink is vannak, akkor a Lagrange-függvényes felírás* $L(x, \alpha, \beta) = f(x) + \sum_{i=1}^{m} \alpha_i c_i(x) + \sum_{i=1}^{p} \beta_j h_j(x)$. α_i*-t Lagrange-multiplikátornak nevezzük, amely az* i*-edik* $c_i x \leq$ *egyenlőtlenségi feltételhez tartozik, hasonlóképpen* β_i *a* $h_j x = 0$ *egyenlőségi feltételhez tartozó Lagrange-multiplikátor. Az* α, β *vektorokat a primál problémához tartozó duális változóknak vagy Lagrange-multiplikátor vektoroknak nevezzük.*

A Lagrange-függvény után jöjjön most az ahhoz tartozó Lagrange-duális.

2.0.6. Definíció. *[3,5.1.2] A Lagrange-duális függvényt úgy definiáljuk, hogy vesszük a Lagrange-függvény minimumát az* x*-ek felett*

$$g(\alpha) := \inf_{x \in \mathbb{R}^n} L(x, \alpha) = \inf_{x \in \mathbb{R}^n} \left(f(x) + \sum_{i=1}^{n} \alpha_i c_i(x) \right)$$

Tehát a duális feladat

$$\arg\max_{\alpha \in \mathbb{R}^n} g(\alpha) \quad \textit{ahol } \alpha \geq 0$$

Ha a Lagrange-függvény korlátlanul csökken x-re nézve, akkor a duális függvény értéke $-\infty$.
Hasonlítsuk össze a primál és duál feladat optimumjait.

2.0.7. Állítás. *[3,5.1.3] Ha $p^\star$ a primál feladat optimuma, míg $d^\star$ a Lagrange-dál feladat optimuma (infimum és szuprémum), akkor $d^\star \leq p^\star$.*

Bizonyítás. [3,5.1.3] Minden $\alpha \geq 0$-ra $g(x, \alpha) \leq d^\star$. Legyen x' megengedett megoldás, vagyis teljesíti a $c_i x \leq 0$ feltételeket, ekkor $\sum_{i=1}^{m} \alpha_i c_i(x') \leq 0$.

$$L(x', \alpha) = f(x') + \sum_{i=1}^{m} \alpha_i c_i(X') \leq f(x')$$

Ebből jön

$$g(\alpha) = \inf_{x \in \mathbb{R}^d} L(x, \alpha) \leq L(x', \alpha) \leq f(x').$$

Minden megengedett x' pontra teljesül $g(\alpha) \leq f(x')$, így $g(\alpha) \leq p^\star$ is. □

A $d^\star \leq p^\star$ egyenlőtlenséget szokták gyenge dualitásnak is nevezni. A dualitási rés $p^\star - d^\star$ különbség, ha ez 0, akkor áll fenn az erős dualitás. Az erős dualitás egyik elégséges feltétele a Slater feltételek.

2.0.8. Állítás. *Slater feltételek [3, 5.26 és 5.27] Legyen $\mathcal{X} \subseteq \mathbb{R}^n$ konvex, valamint $c_1, \ldots, c_m : X \to \mathbb{R}$ konvex függvények. Azt állítjuk ez a két állítás ekvivalens:*

- *$\exists x \in \mathcal{X}$, amelyre $c_i x < 0 \quad \forall i = 1, 2, \ldots, m$-re, vagyis az egyenlőtlenségek szigorúan teljesülnek.*
- *Ha $\alpha \neq 0$, akkor $\forall \alpha \geq 0 \quad \exists x \in \mathcal{X}$ úgy, hogy $\sum_{i=1}^{n} \alpha_i c_i(x) < 0$*

A gyakorlatban legtöbbször differenciálható függvényekkel dolgozunk, így ezek körében keresem az optimalitási feltételeket.

2.0.9. Tétel. *Elégséges Karush-Kuhn-Tucker (KKT) feltételek konvex, differenciálható esetben [3,5.5.3]*
Legyenek $f, c_1, \ldots, c_m : \mathbb{R}^n \to \mathbb{R}$ konvex, differenciálható függvények. Bármely x', α' pontok, amelyek kielégitik az alábbi egyenlőtlenségeket

- $c_i(x') \leq 0, \quad i = 1, \ldots, m$
- $\alpha_i c_i(x') = 0, \quad i = 1, \ldots, m$
- $\alpha_i \geq 0, \quad i = 1, \ldots, m$
- $\partial_x L(x', \alpha') = \partial_x f(x') + \sum_i \alpha_i \partial_x c_i(x') = 0$

ez esetben x' és α' primálisan és duálisan optimálisak.

2.0.10. Definíció. *[6] Legyenek $f, c_1, \ldots, c_n : \mathbb{R}^n \to \mathbb{R}$ konvex, differenciálható függvények. A primál feladhatoz tartozó Wolfe-duális feladatot így definiáljuk*

$$\arg\max_{x,\alpha} L(x,\alpha) = f(x) + \sum_{i=1}^{m} \alpha_i c_i(x)$$

ahol teljesül $\partial_x L(x,\alpha) = \partial_x f(x) + \sum_{i=1}^{m} \alpha_i \partial_x c_i(x) = 0$ és $\alpha_i \geq 0 \quad$ minden $i = 1, \ldots, m$-re.

Tehát ellentétben a Lagrange-duálissal, itt az x is döntési változó. Az egyenlőségi feltétel miatt a Wolfe-duális nem biztos hogy konvex lesz. Ha a primál feladat megoldható, akkor gyenge dualitás áll fenn, ha teljesülnek a Slater-feltételek, akkor pedig erős dualitás teljesül.

A konvex optimalizálás egy szelete a kvadratikus programozás, ilyen feladatokkal fogunk főleg találkozni, így érdemes röviden áttekinteni.

2.0.11. Definíció. *[1,6.72] Minimalizálunk egy x-ben kvadratikus függvényt*

$$\min_x \quad \frac{1}{2} x^T K x + c^T x$$

lineáris feltételek mellett

$$Ax + d \leq 0$$

ahol $x, c \in \mathbb{R}^m$, $d \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times m}$ és $K \in \mathbb{R}^{m \times m}$ egy pozitív definit mátrix.

A primál problémához tartozó Lagrange-függvényes felírás [1,6.73]:

$$L(x,\alpha) = \frac{1}{2} x^T K x + c^T x + \alpha^T (Ax + d)$$

ahol $\alpha \geq 0$. A feltételek és a célfüggvény is mind konvexek, differenciálhatóak, így a KKT-feltételek így alakulnak ebben a speciális esetben:

- $\alpha^T (Ax + d) = 0$
- $Ax + d \leq 0$
- $\alpha \geq 0$
- $\partial_x L(x,\alpha) = Kx + A^T \alpha + c = 0$

Ekkor x, α primálisan és duálisan optimálisak.

Ezekből a feltételekből ki tudjuk hozni a Lagrange-duális alakot is. A 4. megszorításból jön $x = -K^{-1}(A^T\alpha + c)$, mivel K-t tudtuk invertálni a pozitív definitsége miatt. Átalakítunk

$$L(x, \alpha) = \frac{1}{2}x^T Kx + c^T x + \alpha^T(Ax + d) = -\frac{1}{2}\alpha^T AK^{-1}A^T\alpha - (c^T K^{-1}A^T + d)\alpha - \frac{1}{2}c^T K^{-1}c$$

ezt szeretnénk maximalizálni α tekintetében. Mivel az utolsó összeadandó tag α-tól független, így elhagyható. A duális felírás

$$\text{argmax}_\alpha \left(-\frac{1}{2}\alpha^T AK^{-1}A^T\alpha - (c^T K^{-1}A^T + d^T)\alpha \right),$$

ahol $\alpha \geq 0$. Ennek a duális felírásnak az az előnye a primállal szemben, hogy a sovány feltételek jobban kezelhetőek.

3. fejezet

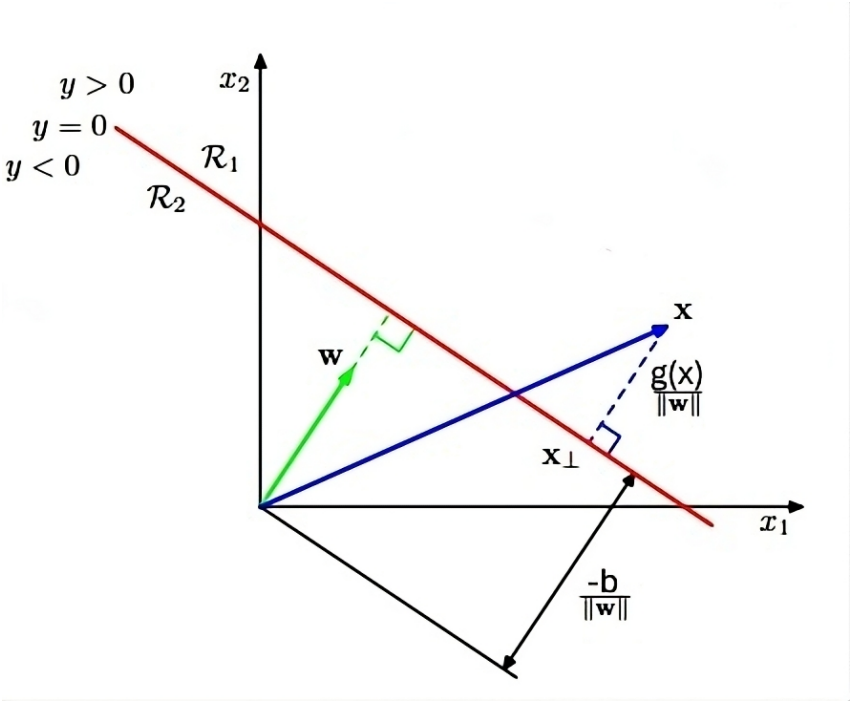
Szupport vektor gépek bevezetése

A szupport vektor gépeket(rövidítve innentől SVM) először a bináris klasszifikáció feladatára fejlesztették ki, majd használukat kiterjesztették a több osztályos és a regressziós problémákra is. Szóval a bináris klasszikációs feladatnál adott egy ismeretlen, $P(x, y)$ eloszlásból generált $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ minta, ahol $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{+1, -1\}$. A hipersíkok $\langle w, x\rangle + b = 0$ alakban felírhatóak, s ezekből képezzük a döntési függvényeinket. A feladat célja, hogy a $y_i = 1$ címkézésű pontok a hipersík azonos oldalára essenek (ugyanez igaz $y_i = -1$ pontokra is). Tehát a pontokat úgy szeparáljuk a hipersíkkal, hogy a $+1$-es és -1-es címkézésű pontok átoldalt legyenek, így $f(x) = \operatorname{sgn}(\langle w, x\rangle + b)$ osztályozók adódnak ebből.

Feltesszük, hogy létezik legalább egy olyan hipersík, amely helyesen választja el egymástól a $y_i = +1$, $y_j = -1$ pontokat. Azt a hipersíkot keressük, amelynek legjobb az általánosító képessége, még nem látott, új x adatpontokat a lehető legjobb osztályozza. A 1.0.12 tétel alapján maximalizálnunk kell a margót, vagyis a hipersíkot úgy kell elhelyeznünk, hogy a hozzá legközelebbi pont távolsága a lehető legnagyobb legyen. Ennek érdekében szükségünk lesz egy pont és egy hipersík távolságának képletére.

3.0.1. Állítás. *[2,7.2] Adott egy $\mathcal{S} = \{\langle w, x\rangle + b = 0 | x \in \mathbb{R}^d\}$ affin hipersík és egy x_n pont távolsága ekképpen számolható:*

$$\operatorname{dist}(x_n, \mathcal{S}) = \frac{|\langle w, x_n\rangle + b|}{\|w\|}$$



3.1. ábra. Hipersík és Pont Távolsága [2, 4.1 Figure)]

Bizonyítás. Ha egy x_A, x_B eleme a hipersíknak, akkor $\langle w, x_A\rangle + b = 0$ és $\langle w, x_B\rangle + b = 0$, így $\langle w, (x_A - x_B)\rangle = 0$. Tehát w ortogonális minden hipersíkbeli vektorra, w normálvektora a hipersíknak. Ahogyan 3.1-n is látható, legyen $x_\perp$ az x_n-nek $\mathcal{S}$-re vett ortogonális projekció által kapott képe. Nevezzük $\text{dist}(x_n, \mathcal{S})$ röviden r-nek. Így felírható

$$x_n = x_\perp + r\frac{w}{\|w\|} \Longrightarrow \langle w, x_n\rangle = \langle w, x_\perp\rangle + r\frac{\|w\|^2}{\|w\|}.$$

Most adjunk hozzá mindkét oldalhoz b-t

$$\langle w, x_n\rangle + b = (\langle w, x_\perp\rangle + b) + r\frac{\|w\|^2}{\|w\|} = r\|w\|.$$

Innen pedig megkapjuk az $r = \frac{|\langle w, x_n\rangle + b|}{\|w\|}$ a keresett összefüggést. □

Mivel $f(x_n) = \text{sgn}(\langle w, x_n\rangle + b) > 0$, ha $y_n = 1$, illetve $f(x_n) = \text{sgn}(\langle w, x_n\rangle + b) < 0$, ha $y_n = -1$. Így $f(x_n)y_n$ mindenképpen pozitív. Legyen $g(x) = \langle w, x\rangle + b$ lineáris diszkriminánsfüggvény-ezt úgy kapjuk, hogy az $f(x)$ osztályozási függvényből elhagyjuk a szignum függvényt, így $g(x)$ értékkészlete folytonos, $g(x_n)y_n$ szintén pozitív lesz $\forall n$-re.

$$r = \frac{|\langle w, x_n\rangle + b|}{\|w\|} = \frac{y_n g(x_n)}{\|w\|} = \frac{y_n(\langle w, x_n\rangle + b)}{\|w\|}$$

A margó a legközelebbi adatpont távolsága a döntési hipersíktól, ezt szeretnénk maximalizálni az optimális w, b paraméterek használatával, amelyek egyértelműen meghatározzák a hipersíkot. Szóval a maximális margó probléma

$$\arg\max_{w,b}\left\{\frac{1}{\|w\|}\min_n\left[y_n(\langle w,x_n\rangle+b)\right]\right\}$$

$\frac{1}{\|w\|}$-t azért hozhattuk ki az n-re történő minimalizálásból, mert nem függ n-től. Ez a feladat viszont elég bonyolult, ezért ezzel ekvivalens könnyebb kihívássá szeretnénk alakítani. Ha átskálázzuk a döntési hipersíkot $w \to kw$, $b \to kb$, akkor sem változik x_n távolsága a hipersíktól, $\frac{y_n(\langle kw,x_n\rangle+kb)}{\|kw\|} = \frac{y_n(\langle w,x_n\rangle+b)}{\|w\|}$. Így nyugodtan választhatjuk a legközelebbi adatpontra ezt a feltételt $y_n(\langle w,x_n\rangle+b)=1$, minden pontra fennáll ez a korlát:

$$y_i(\langle w,x_i\rangle+b)\geq 1 \quad i=1,2,\dots n-re$$

Ez a döntési hipersík kanonikus reprezentációja [2,7.5]. Itt most feltettük, hogy az adatok (lineárisan) szétválaszthatóak (tehát létezik legalább egy hipersík, amely tökéletesen, hiba nélkül elszeparálja egymástól a $+1$ és -1-s címkézésű pontokat), ez adja a kemény margó feladatot. Azokat az adatpontokat, ahol a korlátozás egyenlőséggel teljesül, aktívnak nevezzük őket, a többi pont pedig inaktív. Az aktív pontok a szupport vektorok. A célunk $\frac{1}{\|w\|}$ maximalizálása, ami ekvivalens azzal, ha $\|w\|^2$-t minimalizáljuk.Az optimalizálási probléma eme formája:

$$\arg\min_{w,\,b}\ \frac{1}{2}\|w\|^2$$

Az $\frac{1}{2}$-s szorzót a későbbi számítások kényelmesebbé tétele miatt vettük hozzá. Így kaptunk egy konvex optimalizálási feladatot feltételekkel, pontosabban egy kvadratikus programozási feladatot, mivel egy négyzetes célfüggvényt minimalizálunk lineáris egyenlőtlenségi korlátok mellett. A probléma Lagrange-függvényes felírása

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i\left\{y_i(\langle w,x_i\rangle+b)-1\right\}$$

Kihasználjuk, hogy a feladat differenciálható is, így a KKT-feltételeket alkalmazva $\partial_x L(x,\alpha)=\partial_{w,b}L(w,b,\alpha)=0$. A deriváltak

$$\partial_w L(w,b,\alpha)=w-\sum_{i=1}^{n}\alpha_i y_i x_i=0\longrightarrow w=\sum_{i=1}^{n}\alpha_i y_i x_i$$

$$\partial_b L(w,b,\alpha)=\sum_{i=1}^{n}\alpha_i y_i=0$$

Az $\alpha_i c_i(x) = 0$ feltételekből

$$\alpha_i \left(y_i \left(\langle x_i, w \rangle + b \right) - 1 \right) = 0$$

Vonjunk le a kapott összefüggésekből következtetéseket. A 3. egyenlet a legegyértelműbb, az inaktív pontoknak $\alpha_i = 0$ adódik, $\alpha_i > 0$ értékek csak aktív pontokhoz tartozhatnak. Az 1. egyenletből kapjuk, hogy w előáll az x_i-k lineáris kombinációjaként, azonban az α_i együtthatók miatt ezt tovább szűkíthetjuk az aktív pontokra. Vagyis a döntési hipersík w paramétere meghatározható a hozzá legközelebb, margón lévő pontok segítségével. Ezek a szupport vektorok, elnevezés mögötti szimbolika az, hogy ezek tartják a hipersíkot két oldalról. A 2. egyenletből következik, hogyha az összes x_i-re egy-egy α_i nagyságú erő hatna a hipersíkra merőlegesen (y_i által meghatározott irányban), akkor ezek az erők kiegyenlítenék egymást, eredő erő zérus lenne.

Próbáljuk eliminálni a w, b-t az $L(w, b, \alpha)$ Lagrange-függvényes felírásból, hogy eljussunk a duális problémáig. Használjuk a KKT-feltételekből kapott korlátozásokat.

$$L(w,b,\alpha) = \frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^2 - \sum_{i=1}^{n} \alpha_i \left\{ y_i \left(\sum_{j=1}^{n} \langle x_i, \alpha_j y_j x_j \rangle + b \right) - 1 \right\} =$$

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^{n} \alpha_i \left\{ y_i \left(\sum_{j=1}^{n} \langle x_i, \alpha_j y_j x_j \rangle + b \right) - 1 \right\} =$$

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - b\sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i =$$

$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{i=1}^{n} \alpha_i$$

Mivel $\sum_{i=1}^{n} \alpha_i y_i = 0$, így a $-b\sum_{i=1}^{n} \alpha_i y_i$ kiesett.
Tehát a duális feladatunk [8, 6.6 Proposition]:

$$\underset{\alpha \in \mathbb{R}^n}{\operatorname{argmax}} \quad \tilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{ahol} \quad \alpha_i \geq 0 \quad \text{és} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Kaptunk egy kvadratikus programozási feladatot. Ezek megoldására léteznek hatékony algoritmusok, megoldó programok. Gyakorlatban általában ezeket alkalmazzuk.

A $g(x) = \langle w, x \rangle + b$ lineáris diszkrimináns függvénybe visszahelyettesítve a w-re kapott összefüggést

$$g(x) = \sum_{i=1}^{n} \alpha_i y_i \langle x, x_i \rangle + b.$$

Ezt használjuk az új pontok osztályozására. Azok a pontokra, melyekhez tartozó α_i Lagrange-multiplikátorok 0-k, nem vesznek részt az új pontok osztályozásában, csak a szupport vektorokra hagyatkozunk. Ez is egy nagy előnye az SVM-nek más osztályozókkal szemben, hogy a tesztadatok relatíve kis hányadára hagyatkozik az új minták szelektálásában.

Ha meghatároztuk α-t, akkor w ebből könnyen adódik, szimplán csak behelyettesítünk a w-re kapott összefüggésünkbe. Ahhoz, hogy a hipersík másik paraméterét, b-t megkapjuk, kihasználjuk, hogy minden x_i szupport vektorra igaz $y_i(\langle w, x_i \rangle + b) = 1$, $\alpha_i > 0$. Így a w-re kapott összefüggést erre használva

$$y_i \left(\left\langle x_i, \sum_{j=1}^{n} \alpha_j y_j x_j \right\rangle + b \right) = y_i \left(\sum_{j \in S}^{n} \alpha_j y_j \langle x_i, x_j \rangle + b \right) = 1,$$

ahol S a szupport vektorok indexhalmaza. Mivel $y_i^2 = 1$ mindenképpen, y_i-vel beszorzás után könnyen kifejezhető b ily módon

$$b = y_i - \sum_{j \in S}^{n} \alpha_j y_j \langle x_i, x_j \rangle.$$

Egy önkényesen megválasztott x_i szupport vektor helyett célszerű több pont segítségével kiszámolni a nagyobb numerikus stabilitás érdekében. Numerikusan legstabilabb eredményt úgy érhetjük el, hogy az összes szupport vektorra külön-külön kiszámoljuk az abból adódó b-t, s ezeket átlagoljuk

$$b = \frac{1}{N_S} \sum_{i \in S} \left(y_i - \sum_{j \in S}^{n} \alpha_j y_j \langle x_i, x_j \rangle \right).$$

ahol N_S a szupport vektorok száma.

Eddig azt az esetet láttuk, hogyha a bemeneti térben az adatok lineárisan szeparálhatóak, akkor az SVM-ek jól működő konstukciók. Azonban szeretnénk ennél tovább lépni. Mint a lineáris módszereknél általában, általánosítunk. A bemeneti teret kibővítjuk bázisfüggvények segítségével. Általában igaz, hogy az ilyen kibővített terekben elért lineáris határok jobb szeparációt eredményeznek a tanító halmazunkban, és ezek a döntési határok nemlineáris döntési határoknak felelnek meg az eredeti bemeneti térben.

Vegyük a $h_j(x)$ bázisfüggvényeket, ahol $j = 1, 2 \dots m$. Az $x_1, x_2, \dots x_n$ tanító adatok helyett $h(x_1), h(x_2), \dots h(x_n)$ adatokkal dolgozunk, ahol $h(x_i) = (h_1(x_i), h_2(x_i), \dots, h_m(x_i))$. Az osztályozónk $\text{sgn}(g'(x))$, $g'(x) = w^T h(x) + b$. A $h(x_i)$-k alapján betanított SVM-hez tartozó Lagrange-duális feladat

$$\tilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle h(x_i), h(x_j) \rangle$$

$$g'(x) = w^T h(x) + b = \sum_{i=1}^{n} \alpha_i y_i \langle h(x), h(x_i) \rangle + b$$

A megoldás csak skaláris szorzatokon keresztül tartalmazza $h(x)$-et, tehát nincs szülségünk expliciten kiszámolni a $h(x)$-t minden adatpontra, elegendő ismerni az úgynevezett kernel függvényt

$$K(x_i, x_j) = \langle h(x_i), h(x_j) \rangle$$

Ez a függvény kiszámítja az átalakított térben a skaláris szorzatokat. Azt szeretnénk, hogy megfelelő h választás esetén költséghatékonyan tudjuk kiszámolni a skaláris szorzatokat. Így jutunk el a kernelek a témaköréig.

4. fejezet

Kernelek

Az SVM-eknél a minták egy skalárszorzatos térben léteznek, azonban velük kapcsolatban szeretnénk általánosabb hasonlósági mértéket használni egy leképezes használatával, ami lehet nemlineáris is. Ezért ebben a fejezetben bevezetem a legfontosabb definíciókat, állításokat a kernelek kapcsán annak érdekében, hogy el tudjunk jutni az oly fontos reprezentációs tételig.

4.0.1. Definíció. *[1, 2.1 és 2.2] Adott egy tetszőleges halmaz $\mathcal{X}$ legyen $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ egy kernel függvény $\mathcal{X}$ felett, ha létezik egy olyan $\mathcal{H}$ Hilbert-tér és egy $\Phi : X \to \mathcal{H}$ leképezés, hogy*

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}.$$

Ekkor ezt a Φ-t tulajdonság leképezésnek, azt a $\mathcal{H}$ Hilbert-teret, ahova Φ képez pedig tulajdonságtérnek nevezzük.

A kernel tulajdonképpen egy olyan $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ kétváltozós függvény, amely kettő $\mathcal{X}$-beli ponthoz egy hasonlóságukat jellemző számot rendel.
Pár SVM-eknél is gyakran használt kernel:

- d-ed fokú polinomiális kernel $\quad k(x, x') = (1 + \langle x, x' \rangle)^d$
- Gauss RBF kernel $\quad k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$ $\quad$ ahol $\sigma > 0$ paraméter
- neurális háló kernel $\quad k(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2)$
  ahol κ_1, κ_2 paraméterek beállítják a kernel működését.

4.0.2. Példa. *[4, 12.23 és 12.24] Ha veszünk egy másodfokú polinomális kernelt, akkor ennek egy gyakorlati szemléltetése (adott egy tulajdonságtér x_1 és x_2 bemenettel)*

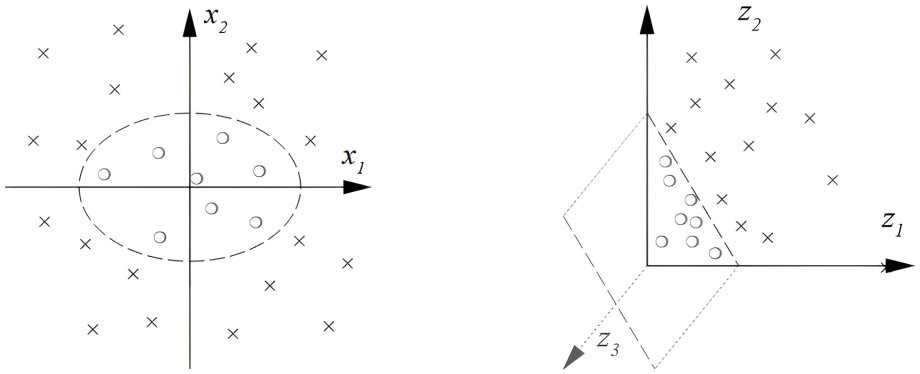
$$k(x,x') = (1+\langle x,x'\rangle)^2 = (1+x_1x_1'+x_2x_2')^2 = 1+2x_1x_1'+2x_2x_2'+(x_1x_1')^2+(x_2x_2')^2+2x_1x_1'x_2x_2'$$

Ekkor $m = 6$, a következő bázisfüggvényeket használva $h_1(x) = 1, h_2(x) = \sqrt{2}x_1, h_3(x) = \sqrt{2}x_2, h_4(x) = X_1^2, h_5(x) = x_2^2, h_6(x) = \sqrt{2}x_1x_2$, s így $k(x,x') = \langle h(x), h(x')\rangle$.
A $h_1(x), h_2(x), \ldots, h_6(x)$ bázisfüggvényekkel képezzük a bemeneti vektort egy magasabb dimenziós térbe, ahol az algoritmusunk könnyebben el tudja választani az adatokat egy lineáris döntési határral. Minden x-ből egy 6 dimenziós vektort képeztünk.

Nem kell expliciten kiszámolnunk $h(x)$-et, a kernel függvény megadja nekünk a skaláris szorzatot ami a feladatban is jól használható

$$\tilde{g}(x) = \sum_{i=1}^{n} \alpha_i y_i \langle h(x), h(x_i)\rangle + b = \sum_{i=1}^{n} \alpha_i y_i k(x, x_i) + b$$

A következő egyszerű szemléltető példában 4.1 az ábra bal oldalán, a bemeneti térben a valós döntési határ egy ellipszis. A pontokat egy nemlineáris leképezéssel a tulajdonságtérbe képezzük, például $\Phi(z_1, z_2, z_3) = (x_1^2, x_2^2, 2x_1x_2)$-vel. Az ellipszis így egy hipersíkká alakul (jobb oldal). A bináris klasszifikációs probléma a tulajdonságtérben a hipersík meghatározására egyszerűsödik az oda képzett adatok alapján.



4.1. ábra. Kernel szemléltetése a tulajdonságtérben [1, 2.1 Figure]

4.0.3. Definíció. *[1, 2.15] A $K \in \mathbb{R}^{n\times n}$ mátrix pozitív szemidefinit, ha teljesül*

$$\langle Kx, x\rangle \geq 0 \quad \textit{minden } x \in \mathbb{R}^n \textit{ esetén.}$$

K pozitív definit, ha

$$\langle Kx, x\rangle > 0 \quad \textit{minden } x \in \mathbb{R}^n \setminus \{0\} \textit{ esetén,}$$

4.0.4. Definíció. *[1, 2.3 Definition, 2.5 Definition] A $\mathcal{X}$ (ami általában $\mathbb{C}$ vagy $\mathbb{R}$) halmazon értelmezett $k : X \times X \to \mathbb{R}$ kernel Gram mátrixa az $x_1, \ldots, x_n \in \mathcal{X}$-nek az a K mátrix, melynek elemei*

$$K_{ij} := k(x_i, x_j).$$

A k kernel pozitív definit, ha a Gram mátrixa minden $x_1, \ldots, x_n \in X$-re pozitív szemidefinit. Hasonlóan, k szigorúan pozitív definit, ha a Gram mátrixa minden $x_1, \ldots, x_n \in X$-re pozitív definit.

A kernel függvények szimmetrikusak a skaláris szorzat szimmetrikussága miatt
$k(x, x') = \langle \Phi(x), \Phi(x')\rangle = \langle \Phi(x'), \Phi(x)\rangle = k(x', x)$.

4.0.5. Állítás. *Cauchy-Schwarz-Bunyakovszkij-egyenlőtlenség a pozitív definit kernelekre [1, 2.7 Proposition] Ha k egy pozitív definit kernel, $x_1, x_2 \in \mathcal{X}$, akkor*

$$|k(x_1, x_2)|^2 \leq k(x_1, x_1)k(x_2, x_2)$$

Erre csak egy szemléletes bizonyítást adunk ([1, 2.7 Proposition]).
$K_{ij} = k(x_i, x_j)$, legyen $i, j \in \{1, 2\}$ Gram-mátrix pozitív szemidefinit, ezért a determinánsa nemnegatív, azaz

$$0 \leq \det(K) = \det \begin{pmatrix} K_{11} & K_{12} \\ K_{12} & K_{22} \end{pmatrix} = K_{11}K_{22} - K_{12}^2.$$

vagyis

$$|K_{12}|^2 \leq K_{11}K_{22}.$$

Ez pedig átírva a keresett egyenlőtlenség

$$|k(x_1, x_2)|^2 \leq k(x_1, x_1)k(x_2, x_2).$$

A pozitív definit kernelekkel szeretnénk dolgozni, ehhez keresünk térstruktúrát, ahol a kernelekkel, mint skaláris szorzat foglalkoznánk. Ehhez alkották meg a reprodukáló magú Hilbert-tereket (Reproducing Kernel Hilbert Space angolul, innen jön az RKHS használatos rövidítés).
$\mathcal{X}$ egy nemüres halmaz, jelöljük $\mathcal{F}(\mathcal{X}, \mathbb{R})$ az $\mathcal{X} \to \mathbb{R}$ függvények $\mathbb{R}$ feletti vektorterét.

4.0.6. Definíció. *[7] Az $\mathcal{F}$ egy normált tér, akkor a $\varphi : \mathcal{F} \to \mathbb{R}$ funkcionál korlátos, ha $\exists\ M > 0$ úgy, hogy*

$$|\varphi(f)| \leq M\|f\| \quad \textit{minden } f \in \mathcal{F}\textit{-re.}$$

4.0.7. Definíció. *[1, 2.9 Definition] $\mathcal{H} \subseteq \mathcal{F}(\mathcal{X}, \mathbb{R})$ egy reprodukáló magú Hilbert-tér, ha ezek a feltételek teljesülnek:*

- *$\mathcal{H}$ egy vektortér*
- *$\mathcal{H}$ Hilbert-teret alkot egy hozzá tartozó $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ skaláris szorzattal*
- *$\forall x \in X$ ponthoz tartozik egy $E_x : H \to \mathbb{R}$ kiértékelési funkcionál, amelyre $E_x(f) := f(x)$ és ez a funkcionál korlátos*

A feltételeknek megfelelő $\mathcal{H}$ egy valós RKHS $\mathcal{X}$ felett.

4.0.8. Definíció. *[1, 2.29 és 2.30] Ha $\mathcal{H}$ egy RKHS $\mathcal{X}$ felett, akkor létezik egy egyértelmű $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ függvény reprodukáló kernel, amelyre*

- $\forall x \in X, \quad k(\cdot, x) \in \mathcal{H}$*;*
- $\forall f \in \mathcal{H},\ \forall x \in X : \quad f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}}$ *(reprodukáló tulajdonság);*
- $\forall x, y \in X : \quad k(x, y) = \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}}$

Van egy f függvényünk az RKHS-ben, ki szeretnénk számolni a függvény értékét egy adott x pontban. Ehhez azonban elég annyi is, hogy a skaláris szorzatot veszünk f-fel és a $k(\cdot, x)$ reprodukáló kernellel.

4.0.9. Állítás. *Kernel-trükk [1, 2.8 Remark] Ha van egy algoritmusunk, ami az $x_1, \ldots, x_n \in \mathcal{X}$ mintát használ a tanuláshoz k pozitív definit kernel használatával, akkor ha lecseréljük k-t egy másik pozitív definit k' kernel függvényre, akkor értelmes algoritmust kapunk.*

A dolog hátterében az áll, hogy az eredeti algoritmus a $\Phi(x_1), \ldots, \Phi(x_n)$ adatokkal dolgozik a Hilbert-térben és itt csak a skaláris szorzást használja. Amikor k-t lecseréljük k'-ra, akkor ugyanaz az algoritmus működik tovább, csak az új, $\Phi'(x_i)$ adatokon.
Általában ezt úgy használják, hogy R^d-s térben kapunk egy algoritmust, majd a kernel trükk segítségével erősebb modelleket építünk. Ez a kernellizálás. Olyan tulajdonságteret is használhatunk, ahol nemcsak, hogy drága lenne kiszámítani a tényleges leképezést, de nem is lehet, például végtelen dimenziós tereknél.

4.0.10. Tétel. *[1,4.2 Theorem] Legyen $\mathcal{X}$ egy tetszőleges halmaz, $\mathcal{H}$ egy valós RKHS $\mathcal{X}$ felett, $\Omega : [0, \infty) \to \mathbb{R}$ egy szigorúan monoton növekvő függvény és $c : (\mathcal{X} \times \mathbb{R} \times \mathbb{R})^d \to \mathbb{R}$ egy tetszőleges veszteségfüggvény. Ekkor a regulárizált kockázati függvény minimumát elérő $f \in H$*

$$\min_{f \in \mathcal{H}} c\Big((x_1, y_1, f(x_1)), \ldots, (x_n, y_n, f(x_n))\Big) + \Omega(\|f\|_{\mathcal{H}})$$

olyan alakban írható fel, hogy:

$$f(x) = \sum_{i=1}^{n} \alpha_i \, k(x_i, x).$$

Vagyis a megoldás a tanító adatokhoz tartozó kernel függvények lineáris kombinációjaként jelenik meg.

Gyakorlatban azonban legtöbbször kibővítjuk konstansokkal a teret a valódi megoldás kereséséhez, a megoldás alakja pedig ilyen

$$f(x) = f_{\mathcal{H}}(x) + b,$$

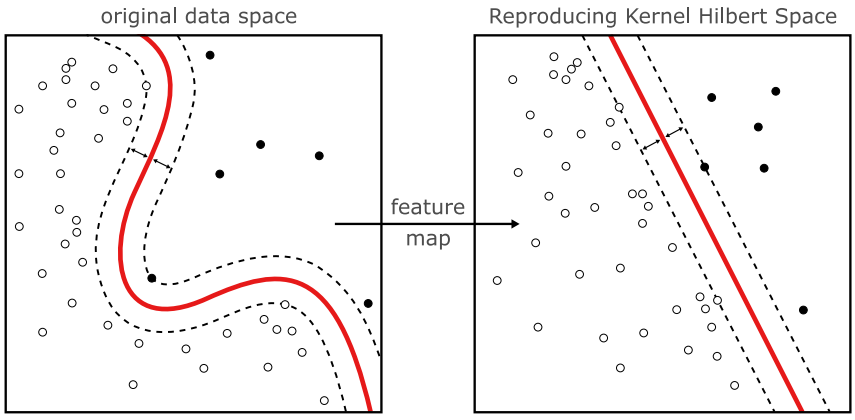
ahol $f_{\mathcal{H}}(x) = \sum_{i=1}^{n} \alpha_i \, k(x_i, x) \in \mathcal{H}$.
Az SVM kernellizált verziója is alkalmazza a reprezentációs tételt a saját igényeihez igazítva, a megoldási alak amivel már talalkoztunk és dolgozni fogunk vele

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i k(x_i, x) + b = \sum_{i=1}^{n} \tilde{\alpha}_i k(x_i, x) + b.$$

5. fejezet

Szupport vektor gépek típusai

Az SVM-mel eddig csak lineáris osztályozást tudtunk végezni azonban a kernellizálás nyomán mostmár tudjuk a nemlineáris problémákat is kezelni. Ahogy a szemléltető ábrán 5.1 is látható, a kernellizálás során a bemeneti adatokat egy magasabb dimenziós térbe képezzük, ahol a tulajdonságtérben az eredeti térben nem lineárisan osztályozható feladat lineárisan osztályozhatóvá válik. Tehát nemlineáris problémát oldunk meg lineáris módszerekkel.



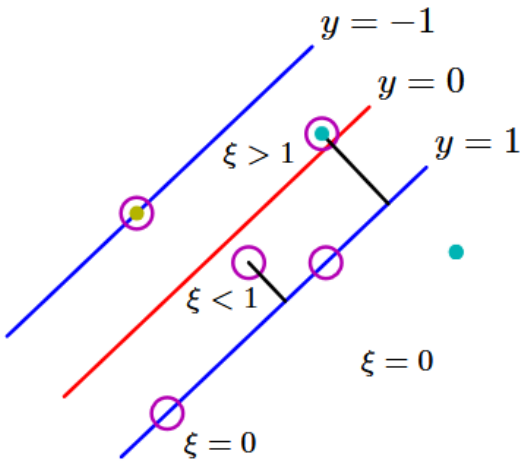
5.1. ábra. Kernellizálás[6]

5.1. C-SVM

SVM-eknél azzal esettel foglalkoztunk, mikor az adatok tökéletesen szeparálhatóak. A tanító adatok a tulajdonságtérben lineárisan elválaszthatóak, ebből az SVM pontos elválasztást ad a bemeneti térben, mégha olykor ez nemlineáris is lesz. Ez azonban in-

kább elméleti jelentőségű eset, mert a valóságban legtöbbször nem ez az ideális eset áll elő. A gyakorlati példákban gyakran átfednek az osztályok a tulajdonságtérben, ekkor a tanító adatok tökéletes elválasztására törekvés gyenge általánoításhoz vezet. Szóval megengedjük, hogy bizonyos pontokat rossz osztályba soroljunk be. A pontok megengedetten a margó rossz oldalára is eshetnek, de büntetjük őket, amely a döntési határtól való távolsággal arányosan nő.

Bevezetünk segédváltozókat, s mint 5.2 is mutatja, $\xi_i \geq 0$, ahol $i = 1, 2, \ldots, n$ minden adatponthoz rendeltünk egyet. Ha egy pont a margó jó oldalán vagy a margón helyezkedik el, akkor $\xi_i = 0$, különben $\xi_i = |y_i - (\langle \Phi(x_i), w \rangle + b)|$. A döntési határon lévő pontokra $g(x_i) = \langle \Phi(x_i), w \rangle + b = 0$, vagyis $\xi_i = 0$, a határ rossz oldalán lévő pontokra már $\xi_i > 0$ lesz.



5.2. ábra. Gyenge margó segédváltozók [2, 7.3 Figure]

Szóval a feladatunk a margó maximalizálása mellett a helytelen osztályozás minimalizálása. Így jutunk el a gyenge margó feladatig, ami a kemény margó feladat relaxációja. Ha minden pontra $\xi_i = 0$, akkor a kemény margó feladatot kapjuk vissza. A gyenge margó primál feladata [8, 6.3]

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{ahol} \quad & y_i(\langle w, \Phi(x_i)\rangle + b) \geq 1 - \xi_i, \quad \forall i = 1, 2, \ldots n \\ & \xi_i \geq 0, \quad \forall i = 1, 2, \ldots n. \end{aligned}$$

A $C > 0$ paraméter (melyről a konstrukció a nevét kapta) hibák súlyának és a marginnak az egyensúlyát szabályozza. Ha $C \to \infty$, akkor szintén visszakapjuk a kemény margóhoz tartozó SVM konstrukciót. A primál feladat Lagrange-függvényes felírása

$$L(w, b, \xi, \alpha, \mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left\{y_i(\langle w, \Phi(x_i)\rangle + b) - 1 + \xi_i\right\} - \sum_{i=1}^{n}\mu_i\xi_i,$$

ahol $\xi_i \geq 0$, $\mu_i \geq 0$ Lagrange-multiplikátorok. Az elégséges KKT-feltételek

$$\begin{cases} y_i(\langle w, \Phi(x_i)\rangle + b) - 1 + \xi_i \geq 0 \\ \xi_i \geq 0 \end{cases} \qquad \text{(primál megengedhetőség)}$$

$$\begin{cases} \alpha_i \geq 0 \\ \mu_i \geq 0 \end{cases} \qquad \text{(duál megengedhetőség)}$$

$$\begin{cases} \alpha_i\big(y_i(\langle w, \Phi(x_i)\rangle + b) - 1 + \xi_i\big) = 0 \\ \mu_i\xi_i = 0 \end{cases} \qquad \text{(komplementeritási feltételek)}$$

ahol $i = 1, 2, \ldots n$. Az optimumnál a Lagrange függvény deriváltja 0 a primál változókra nézve

$$\partial L_{w,b,\xi}(w, b, \xi, \alpha, \mu) = 0.$$

A deriváltak egyesével

$$\begin{aligned} \frac{\partial L}{\partial w} = 0 \quad &\Rightarrow \quad w = \sum_{i=1}^{n}\alpha_i y_i \Phi(x_i) \\ \frac{\partial L}{\partial b} = 0 \quad &\Rightarrow \quad \sum_{i=1}^{n}\alpha_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} = 0 \quad &\Rightarrow \quad \alpha_i = C - \mu_i. \end{aligned}$$

Küszöböljük ki a primál változókat

$$\begin{aligned} L(w, b, \xi, \alpha, \mu) &= \frac{1}{2}\left(\sum_{i=1}^{n}\alpha_i y_i \Phi(x_i)\right)^2 - \sum_{i=1}^{n}\alpha_i\left\{y_i\left(\sum_{j=1}^{n}\langle\Phi(x_i), \alpha_j y_j \Phi(x_j)\rangle + b\right) - 1 + \xi_i\right\} \\ &\quad + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}(C - \alpha_i)\xi_i \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(x_i, x_j) + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i y_i b + \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\alpha_i\xi_i \end{aligned}$$

$$
\begin{aligned}
&- \sum_{i=1}^{n} \alpha_i y_i \left(\sum_{j=1}^{n} \langle \Phi(x_i), \alpha_j y_j \Phi(x_j) \rangle \right) - \sum_{i=1}^{n} C\xi_i + \sum_{i=1}^{n} \alpha_i \xi_i \\
&= -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) + \sum_{i=1}^{n} \alpha_i .
\end{aligned}
$$

Ugyanazt az alakot kaptuk, mint a kemény margó esetében de más feltételekkel. A Wolfe-duális felírás

$$
\begin{aligned}
\underset{\alpha \in \mathbb{R}^n}{\operatorname{argmax}} \quad & \tilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\
& \text{ahol } 0 \leq \alpha_i \leq C \quad \text{és} \quad \sum_{i=1}^{n} \alpha_i y_i = 0.
\end{aligned}
$$

Ez a típusa az SVM-eknek a C-SVM. Ha $\alpha_i \leq C$, akkor $\mu_i > 0$ és a komplementeritási feltételből $\xi_i = 0$, ezek a pontok lehetnek a szupport vektorok. A w, b paraméterek, s ezáltal a hipersík, hasonlóképpen meghatározható itt is, mint a kemény margó feladatnál. A C egy konstans, amely a hibák minimalizálása és a margin maximalizálása közötti kompromisszumot határozza meg. A C önmagában nehezen értelmezhető paraméter, nincs előre ismert módszer a kiválasztására, általában keresztvalidációval becsüljük az optimális értékét. Ezért egy módosítást teszünk a paraméter terén, hogy intuitívabb képet kapjunk a paraméterről, a C-t ν-re cseréljük.

5.1.1. Megjegyzés. *A C-SVM primál feladat sok esetben a hibákat átlagolva írják fel*

$$
\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i
$$

az eredményeken annyit változtat ez, hogy a Wolfe-duális feladat egyik feltétele kissé módosul $0 \leq \alpha_i \leq \frac{C}{n}$, $\frac{1}{n}$*-nel skálázódik, de ez semmi lényeges eredményen változtat érdemlegeset.*

5.2. ν-SVM

A C-SVM esetében C meghatározása rendkívül körülményes tud lenni, esetenként azt a tartományt is nehéz meghatározni, amiben keressük C optimális értékét (pl lehetséges, hogy C=0,001 és C=10000 is ugyanúgy szóba jöhet). Az SVM teljesítményét befolyásoló fő paramétert bizonyos SVM-hez tartozó tulajdonságokhoz szeretnénk kötni, így jött létre a ν-SVM, a típust irányító ν paraméterrel.

A ν-SVM-hez tartozó primál optimalizálási probléma [1, 7.44]

$$\begin{aligned} \min_{w,b,\xi,\rho} \quad & \frac{1}{2}\|w\|^2 - \nu\rho + \frac{1}{n}\sum_{i=1}^{n}\xi_i \\ \text{ahol} \quad & y_i(\langle w, \Phi(x_i)\rangle + b) \geq \rho - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \quad \rho \geq 0, \quad i = 1, \ldots, n. \end{aligned}$$

Bevezetésre került a ν paraméter, valamint egy további optimalizálandó változó, ρ. A korábban alkalmazott a legközelebbi adatpontokra $y_i(\langle w, \Phi(x_i)\rangle + b) = 1$ feltételt elengedjük, ehelyett $y_i(\langle w, \Phi(x_i)\rangle + b) = \rho$ a legközelebbi pont(ok)ra. Mivel megengedjük most is a félreosztályzás lehetőségét, innen jött, hogy minden pontra $y_i(\langle w, \Phi(x_i)\rangle + b) \geq \rho - \xi_i$. Az osztályokat a margók által $\frac{2\rho}{\|w\|}$ távolság választja el egymástól.

5.2.1. Definíció. *[1, 7.43] Vezessük be a margóhiba fogalmát. Olyan pontokat sorolunk ide, amelyekre $\xi_i > 0$, ide tartoznak a jó oldalon lévő, de margón belül lévő és a félreosztályozott pontok. Ezek hányadát az összes pontot tekintve az adott hipersíkhoz tartozóan ez adja meg*

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(y_i(\langle w, \Phi(x_i)\rangle + b) < \rho).$$

5.2.2. Állítás. *[1, 7.5 Proposition] Tegyük fel a ν-SVM eredménye $\rho > 0$. Ekkor*

- *ν felső korlátja a marginhibák arányának*
- *ν alsó korlátja a szupport vektorok arányának.*

Ezzel, hogy szemléletes jelentést kaptunk ν paraméter szerepéről, vezessük le a duális feladatot. A primál feladat Lagrange-függvényes reprezentációja

$$L(w, b, \rho, \xi, \alpha, \mu, \eta) = \frac{1}{2}\|w\|^2 - \nu\rho + \frac{1}{n}\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(\rho - y_i(\langle w, \Phi(x_i)\rangle + b) - \xi_i) - \sum_{i=1}^{n}\mu_i\xi_i - \eta\rho$$

ahol $\alpha_i, \mu_i, \eta \geq 0$ Lagrange-multiplikátorok. A vonatkozó KKT-feltételek

$$\begin{cases} y_i(\langle w, \Phi(x_i)\rangle + b) - 1 + \xi_i \geq 0 & \\ \xi_i \geq 0 & \text{(primál megengedhetőség)} \\ \rho \geq 0 & \end{cases}$$

$$\begin{cases} \alpha_i \geq 0 \\ \mu_i \geq 0 \\ \eta \geq 0 \end{cases} \qquad \text{(duál megengedhetőség)}$$

$$\begin{cases} \alpha_i\big(y_i(\langle w, \Phi(x_i)\rangle + b) - \rho + \xi_i\big) = 0 \\ \mu_i \cdot \xi_i = 0 \\ \eta \cdot \rho = 0 \end{cases} \qquad \text{(komplementeritási feltételek)}$$

Maradtak még a stacionaritási feltételek, a Lagrange-függvény deriváltjai a primál változókra nézve egyesével

$$\frac{\partial L}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{n} \alpha_i y_i \Phi(x_i)$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow \quad \alpha_i = \frac{1}{n} - \mu_i$$

$$\frac{\partial L}{\partial \rho} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i = \nu + \eta.$$

Eliminálva a primál változókat helyettesítésekkel

$$\begin{aligned} L(w, b, \xi, \rho, \alpha, \mu, \eta) &= \frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i \Phi(x_i)\right)^2 + \frac{1}{n}\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \mu_i \xi_i - \nu\rho - \eta\rho \\ &\quad + \sum_{i=1}^{n} \alpha_i \left\{ y_i \left(\sum_{j=1}^{n} \langle \Phi(x_i), \alpha_j y_j \Phi(x_j)\rangle + b \right) - \rho + \xi_i \right\} \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) + \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} \alpha_i \rho \\ &\quad - \sum_{i=1}^{n} \alpha_i y_i \left(\sum_{j=1}^{n} \langle \Phi(x_i), \alpha_j y_j \Phi(x_j)\rangle \right) - \sum_{i=1}^{n} \alpha_i y_i b \\ &\quad + \sum_{i=1}^{n} \alpha_i \rho - \sum_{i=1}^{n} \alpha_i \xi_i \end{aligned}$$

$$= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(x_i, x_j)$$

Így a Lagrange-duális feladat [2, 7.38, 7.39, 7.40, 7.41]

$$\underset{\alpha\in\mathbb{R}^n}{\operatorname{argmax}} \quad \tilde{L}(\alpha) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(x_i, x_j)$$

$$\text{ahol} \quad 0 \le \alpha_i \le \frac{1}{n}, \quad \sum_{i=1}^{n}\alpha_i \le \nu \quad \text{és} \quad \sum_{i=1}^{n}\alpha_i y_i = 0 \quad \forall i = 1, \ldots, n.$$

Ugyanaz a döntési függvény adódik, mint az előbbiekben

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i k(x, x_i) + b\right)$$

A duális feladatot összevetve a C-SVM-nel két fő különbség van. A célfüggvényben nem szerepel $\sum_{i=1}^{n}\alpha_i$ és van egy további megszorításunk is, ami a ν paraméterből adódik, $\sum_{i=1}^{n}\alpha_i \ge \nu$. Emiatt a duális célfüggvény kvadratikusan homogén α-ban, vagyis $\tilde{L}(\lambda\alpha) = \lambda^2\tilde{L}(\alpha)$, ahol $\lambda > 0$ skalár. Ez a feltételek fontosságát hangsúlyozza.

5.2.3. Állítás. *[1,7.6 Proposition] Ha a ν-SVM olyan döntési függvényt eredményez, amelyre $\rho > 0$, akkor ugyanazt a döntési függvényt kapjuk C-SVM esetében is, ha $C = \frac{1}{\rho}$ a választásunk.*

Bizonyítás. Vegyük először a ν-SVM optimalizálási feladatát, rögzítsük le a ρ értékét, ekkor a célfüggvényből a $\nu\rho$ tag elhagyható, mert konstans, így a minimalizálás szempontjából lényegtelen. Az így nyert probléma olyan, mintha a C-SVM-pel dolgoznánk C=1 és $y_i(\langle w, \Phi(x_i)\rangle + b) \ge \rho - \xi_i$ feltétellel, ez ami más, mint az eddigiek. Tegyük fel, hogy ennek a feladatnak a megoldását, vagyis a minimumot w_0, b_0, ξ_0 ismerjük. Ahhoz, hogy a $y_i(\langle w, \Phi(x_i)\rangle + b) \ge 1 - \xi_i$ feltétel teljesüljön, skálázzuk át a változókat, legyen $w' = \frac{w}{\rho}$, $b' = \frac{b}{\rho}$, $\xi' = \frac{\xi}{\rho}$. Az így nyert célfüggvény

$$\frac{1}{2}\rho^2\|w'\|^2 + \frac{\rho}{n}\sum_{i=1}^{n}\xi_i'.$$

Ez a C-SVM feladatát adja vissza $C = \frac{1}{\rho}$-val és egy ρ^2 konstans szorzótényezővel megtoldva, ami az optimalizálás szempontjából ugyanazt a feladatot eredményezi. □

5.3. LS-SVM

A szupport vektor gépek egy újabb fajtája a legkisebb négyzetek módszere (vagyis least squares) után kapta a nevét. Az eddig látott, hagyományos SVM-ek tanítása kvadratikus programozási feladat (a célfüggvény kvadratikus, feltételek pedig lineáris egyenlőtlenségek) megoldását igényli. Az ötlet itt az, hogy egyszerűsítsük a folyamatot azzal, hogy az egyenlőtlenségi korlátokat egyenlőséggé alakítsuk. Így a feltételek lineáris egyenletekké alakulnak. Az LS-SVM-hez tartozó optimalizálási probléma [5, 4.1]

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2$$
$$\text{ahol} \quad y_i(\langle w, \Phi(x_i)\rangle + b) = 1 - \xi_i, \quad i = 1, \ldots, n.$$

A félreosztályozást szokásosan megengedjük, innen ξ_i-k a szokásos segédváltozók. Ha $\xi_i \geq 1$ az x_i hibásan van osztályozva, $\xi_i < 1$ esetven viszont helyesen, de $0 < \xi_i < 1$ esetben margón belül. Az eddigi SVM-ektől eltérően ξ_i lehet negatív is. A primál feladat Lagrange-függvényes felírása

$$L(w, b, \xi, \alpha) = \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 - \sum_{i=1}^{n}\alpha_i(y_i(\langle w, \Phi(x_i)\rangle + b) - 1 + \xi_i),$$

ahol α_i Lagrange-multiplikátorok. Nincs korlátozva nemnegatív értékekre, mint az eddigiekben, mert egyenlőségi megszorításokat használunk. A vonatkozó KKT-feltételek

$$\left\{ \; y_i(\langle w, \Phi(x_i)\rangle + b) = 1 - \xi_i \qquad \text{(primál megengedhetőség)} \right.$$

$$\left\{ \begin{array}{lll} \frac{\partial L}{\partial w} = 0 & \Rightarrow & w = \sum_{i=1}^{n}\alpha_i y_i \Phi(x_i) \\ \frac{\partial L}{\partial b} = 0 & \Rightarrow & \sum_{i=1}^{n}\alpha_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} = 0 & \Rightarrow & \alpha_i = C\xi_i \\ \frac{\partial L}{\partial \alpha_i} = 0 & \Rightarrow & y_i(\langle w, \Phi(x_i)\rangle + b) = 1 - \xi_i. \end{array} \right. \qquad \text{(stacionaritási feltételek)}$$

Mivel egyenletekkel dolgozunk a feltételekben, így nincsenek komplementeritási feltételek. Dolgozzunk az így nyert optimalitási feltételekkel, helyettesítsük be a 4. egyenletbe az 1. és 3. feltétel által nyert kifejezéseket

$$y_i\left(\left\langle \sum_{i=1}^{n}\alpha_i y_i \Phi(x_i), \Phi(x_i)\right\rangle + b\right) = 1 - \frac{\alpha_i}{C}$$

$$\sum_{j=1}^{n} \alpha_j y_i y_j k(x_i, x_j) + y_i b + \frac{\alpha_i}{C} = 1.$$

Definájuk az $\Omega \in \mathbb{R}^{n \times n}$ mátrixot

$$\Omega_{ij} = y_i y_j k(x_i, x_j) + C\,\delta_{ij},$$

ahol δ_{ij} a Kronecker-delta, $\delta_{ij} = 1$, ha $i = j$, különben 0. Az $i = 1, 2, \ldots, n$-hez tartozó egyenleteket együtt vehetjük mátrixos felírással

$$\Omega\alpha + yb = 1.$$

És vegyük még ehhez a fel nem használt 2. feltételt, ami $y^T\alpha = 0$ egy alternatív alakban. A 2 egyenletet összevéve

$$\begin{pmatrix} \Omega & y \\ y^T & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Ez az [5, 4.8] eredménye. A minimalizálási probléma az egyenletrendszer α és b szerinti megoldásával oldható meg. Az Ω mátrix diagonális elemeiben $\frac{1}{C} > 0$, ezért Ω pozitív definit, invertálható lesz így, s ebből

$$\alpha = \Omega^{-1}(1 - yb).$$

Ezt behelyettesítve

$$y^T\alpha = y^T(\Omega^{-1}(1 - yb)) = y^T\Omega^{-1}1 - by^T\Omega^{-1}y = 0 \longrightarrow b = \frac{y^T\Omega^{-1}1}{y^T\Omega^{-1}y}$$

Ennél a típusnál az SVM tanítása kvadratikus programozási feladat helyett lineáris egyenletrendszer megoldására egyszerűsödik. Az LS-SVM-ben minden tanítóadat szupport vektorrá válik (mert minden pontra egyenlőtlenségi feltétel helyett egyenlőség teljesül), így sok adatpontnál ez lassúvá teheti a klasszifikációt. Ennek orvosolására számtalan gyorsítási technika létezik.

5.4. LP-SVM

A legelső SVM-ek esetében kvadratikus programozási feladatokat kellett megoldani. Ha azonban a kvadratikus célfüggvényt egy lineáris függvénnyel helyettesítjük, átalakíthatjuk a problémát egy lineáris programozási feladattá.

Az eddig látott SVM modelleknél L_2 normát használtunk a célfüggvényben, vagyis

$\|w\|_2^2 = w_1^2 + w_2^2 + \dots w_d^2$ volt, amit használtunk.
Ha azonban az L_2 normát L_1 normára cseréljük, ezzel $\|w\|_1 = |w_1| + |w_2| + \dots + |w_d|$, akkor megkapjuk az LP-SVM feladatot [8, 6.1.3]

$$\begin{aligned} \min_{w,b,\xi} \quad & \sum_{i=1}^{d} |w_i| + C \sum_{i=1}^{n} \xi_i \\ \text{ahol} \quad & y_i(\langle w, \Phi(x_i)\rangle + b) \geq 1 - \xi_i, \quad \forall i = 1, 2, \dots n \\ & \xi_i \geq 0, \quad \forall i = 1, 2, \dots n \end{aligned}$$

A reprezentációs tételből a döntési függvény

$$g(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) + b$$

ahol most $\alpha_i \in \mathbb{R}$. Mivel α_i-t valósnak választottuk, nincs szükség y_i-vel való szorzásra, ahogy az előző esetekben, ahol $\alpha \geq 0$ korlátozással éltünk. A döntési függvény általános alakja $\langle w, \Phi(x)\rangle + b$, így visszakövetkeztetve

$$w = \sum_{i=1}^{n} \alpha_i \Phi(x_i).$$

Ebből az az ötletünk támad ahelyett, hogy közvetlenül w-t minimalizálnánk, az α_i-k abszolút értékeit minimalizáljuk. A feladat átirata [5, 4.46]

$$\begin{aligned} \min_{\alpha,b,\xi} \quad & \sum_{i=1}^{d} |\alpha_i| + C \sum_{i=1}^{n} \xi_i \\ \text{ahol} \quad & y_i(\sum_{i=1}^{n} \alpha_i k(x_i, x) + b) \geq 1 - \xi_i, \quad \forall i = 1, 2, \dots n \\ & \xi_i \geq 0, \quad \forall i = 1, 2, \dots n. \end{aligned}$$

Azért, hogy a problémát lineáris programozással meg tudjuk oldani, be kell vezetni $\alpha_i^+, \alpha_i^-, b^-, b^+ \geq 0$ új változókat úgy, hogy $\alpha_i = \alpha_i^+ - \alpha_i^-$ és $b = b^+ - b^-$. Így már megtalálható lineáris programozással α, b és ξ értéke is.

Ebben az átiratban $3n + 2$ változónk van és n db egyenlőtlenségi feltételünk. Nagy adahalmaz esetén még lineáris programozással is lassúvá válhat a tanítás, ezért van szükség különféle dekompozíciós technikákra gyorsítás céljából.

Az LP-SVM célja inkább az, hogy a döntési függvényben kevesebb adatpont vegyen részt a döntésben (egy cél, hogy sok $\alpha_i = 0$ legyen), nem az, hogy a marginon legyenek a szupport vektorok, azok bárhol lehetnek. Az L_1 norma használata mögötti fő törekvés az volt, hogy ritkásabb megoldásokat kapjunk.

6. fejezet

Több osztályos SVM-ek

Az SVM-eket alapvetően bináris klasszifikációra fejlesztették ki, azonban a gyakorlatban jellemző feladat, hogy nemcsak 2, hanem akár jóval több osztályba szeretnénk besorolni a pontokat. Bemutatok néhány módszert, amellyel az eddig látott két osztályos SVM-ekből hatékony több osztályos osztályozót tudunk építeni. Az osztályok számát jelöljük K-val.

6.1. One-versus-the rest avagy egy a többi ellen

K db különálló SVM-et építünk úgy, hogy az n. SVM tanítása során az n. osztály elemeit tekinti pozitívnak, a maradék $K-1$ osztályhoz tartozó adatpontokat pedig negatívnak. Az SVM-ekhez tartozó döntési függvények $f_1, f_2, \ldots, f_k$, ahol $f_i = \operatorname{sgn}(g_i(x))$. Mivel az SVM-ek külön-külön osztályoznak, így előfordulhat, hogy a különböző modellek nem ugyanabba az osztályba sorolják be az új adatpontot. Ennek elkerülése érdekében osztályozáskor azt az osztályt választjuk, melyre [2, 7.49]

$$\arg \max_{i=1,\ldots,K} g_i(x).$$

A $g_i(x)$ értékek segítségével hozhatunk nem besorolási döntéseket is. Például ha a 2 legnagyobb $g_i(x)$ érték közti különbség kisebb egy küszöbértéknél, akkor nem döntünk. Ilyenkor a maradék mintán alacsonyabb hibaszázalék érhető el. A nem besorolt pontok hányadát jegyezzük, ezt jellemzi, hány százalékát kell elutasítani az adatoknak, hogy ez bizonyos hibaszázalékot elérjünk. A módszer egyik hátránya, hogy sok osztálynál erősen kiegyensúlyozatlan adatokkal kell dolgoznunk, például 100 osztálynál csak az adatok 1 százaléka lesz pozitív. A kiegyensúlyozatlanságra egy megoldás [2, 7.1.3] az osztályok címkéinek

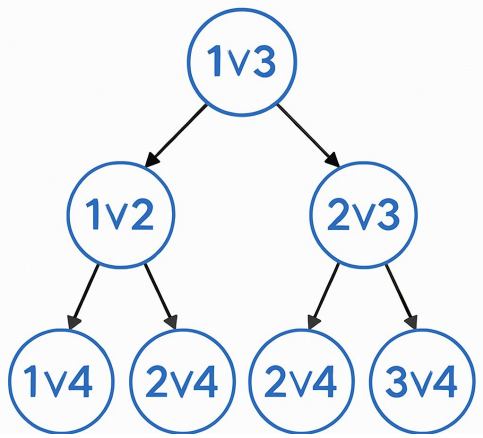
módosítása a tanítás során. A pozitív osztály címkéje maradjon 1, a negatív osztályok címkéi legyenek $-\frac{1}{K-1}$.

6.2. One-versus-one avagy páronkénti osztályozás

Ebben a megközelítésben minden osztálypárra tanítunk egy SVM-et, ez K osztály esetén $\frac{K(K-2)}{2}$ osztályozó betanítását jelenti [1, 7.6.2]. Egy új adatpont osztályba sorolását úgy döntjük el, hogy minden bináris osztályozót kiértékelünk és többségi szavazással döntünk arról, melyik osztály lesz a győztes. Minden SVM dönt arról, hogy a saját 2 osztálya közül melyikbe tartozik az új adatpont-ez a szavazás. A besorolandó osztály az lesz, amelyre a legtöbb szavazat érkezett (döntetlen esetén valamiféle tie-break alkalmazandó). Mivel nagy K esetén sok SVM-et kell tanítani, ez elég idő és számításigényes feladat. Sok esetben már hamar rájövünk bizonyos osztályok nem győzhetnek, ezért egyszerűen felesleges minden SVM-et betanítani. Ennek a módszernek egy továbbfejlesztése a következő konstrukció.

6.3. Irányított aciklikus gráf SVM (DAGSVM)

Az előbb látott $\frac{K(K-2)}{2}$ db SVM-et egy irányított aciklikus gráfba rendezzük, ahol a csúcsok a bináris (1v1) SVM-ek. Az SVM-ek speciális elrendezésével az a célünk ne kelljen az összeset kiértékelni. Egy csúcsban az SVM eldönti, hogy az új bemenet melyik osztályban lesz benne (mégha ez a végeredmény szempontjából nem is lesz feltétlen olyan hasznos), s aszerint halad tovább melyik osztály győzött, az élek a döntési eredménytől függenek. A vesztes osztályt kizárjuk a versenyből, nem lesz az új bemenet osztálya. Minden egyes lépésben egy osztályt eliminálunk a lehetséges osztályokból az új adatpont számára, így aciklisság esetén $K-1$ lépés, ezáltal ugyanennyi SVM kiértékelése elég lesz. 4 osztály esetén egy egyszerű reprezentáció a döntések sorozatának a 6.1 ábra.



6.1. ábra. DAGSVM reprezentáció 4 osztálynál

Az osztályok kiértékelési sorrendje nem befolyásolja a végeredményt, ez csak egy szemléltetés volt egy lehetséges kiinduló SVM esetén. Mivel csak $K - 1$ SVM-et kell kiértékelni, így jóval gyorsabb a sima one-versus-one megközelítésnél.

6.4. Több osztályos SVM-ek összegezve

A több osztályos SVM megközelítések terén nincs egyértelmű favorit, mindig az adott feladat körülményeitől (mintaszám, időkorlát, számítási erőforrás határa) függ, melyik alkalmazható legjobban. Gyakorlatban mégis a one-versus-the rest megközelítés a leggyakoribb, annak ellenére hogy ismertek a korlátai, mert egyszerű és könnyen implementálható.

7. fejezet

A típusok összehasonlítása gyakorlati példákon keresztül

Most pedig nézzünk rá pár olyan feladatra, ahol gyakorlati szempontból átvehetjük az elméletileg áttekintett SVM-típusokat. Megvizsgáljuk az egyes típusok eltérő viselkedését, különböző eredményeit a feladatok kapcsán.

7.1. C-SVM és ν-SVM

Korábban láttuk, hogy bizonyos paraméter választással a 2 típus döntési függvényei megfeleltethetőek egymásnak. Hasonlítsük összse teljesítményüket példákon keresztül. Többféle elemszámú adathalmazt vizsgálunk, $n = 100, 1000, 2000$ eseteket, az adatunk alapvetően normál generálású, de az osztályok el vannak tolva, valamint Gauss-zaj is hozzá van adva, így értelemszerűen tökéletes szeparációt nem is lehet elérni. E két SVM sajátosságai alapvetően meghatározzák a C és ν paraméterek. Nagy C választása esetén az SVM szigorúan kezeli az osztályozási hibákat, jobban célja a tanító halmazon nagyobb pontosság elérése, mint alacsony C választás esetén, melynél a jobb általánosítás fontos-ez zajos adatoknál előnyt jelenthet. A ν intuitívabb jelentése hasznunkra van azzal, hogy a ν felső korlátot ad a margin hibák arányára és alsó korlátot ad a szupport vektorok arányára. A valós adatok azonban nem tökéletesen szeparálhatóak, ezért a hibaarány ν alatti értékre való csökkentése nem mindig lehetséges.

== 100 minta ==

C	Osztályozási hiba (C-SVM)	SV arány (C-SVM)	nu	Osztályozási hiba (Nu-SVM)	SV arány (Nu-SVM)
1	0.18	0.59	0.2	0.10	0.38
10	0.16	0.45	0.5	0.18	0.56
100	0.10	0.39	0.7	0.20	0.74
1000	0.10	0.34	0.9	0.19	0.92

== 1000 minta ==

C	Osztályozási hiba (C-SVM)	SV arány (C-SVM)	nu	Osztályozási hiba (Nu-SVM)	SV arány (Nu-SVM)
1	0.216	0.475	0.2	0.232	0.340
10	0.216	0.458	0.5	0.214	0.506
100	0.207	0.448	0.7	0.213	0.704
1000	0.204	0.447	0.9	0.214	0.902

== 2000 minta ==

C	Osztályozási hiba (C-SVM)	SV arány (C-SVM)	nu	Osztályozási hiba (Nu-SVM)	SV arány (Nu-SVM)
1	0.2275	0.491	0.2	0.6870	0.3235
10	0.2240	0.475	0.5	0.2325	0.5045
100	0.2145	0.469	0.7	0.2345	0.7015
1000	0.2135	0.462	0.9	0.2355	0.9010

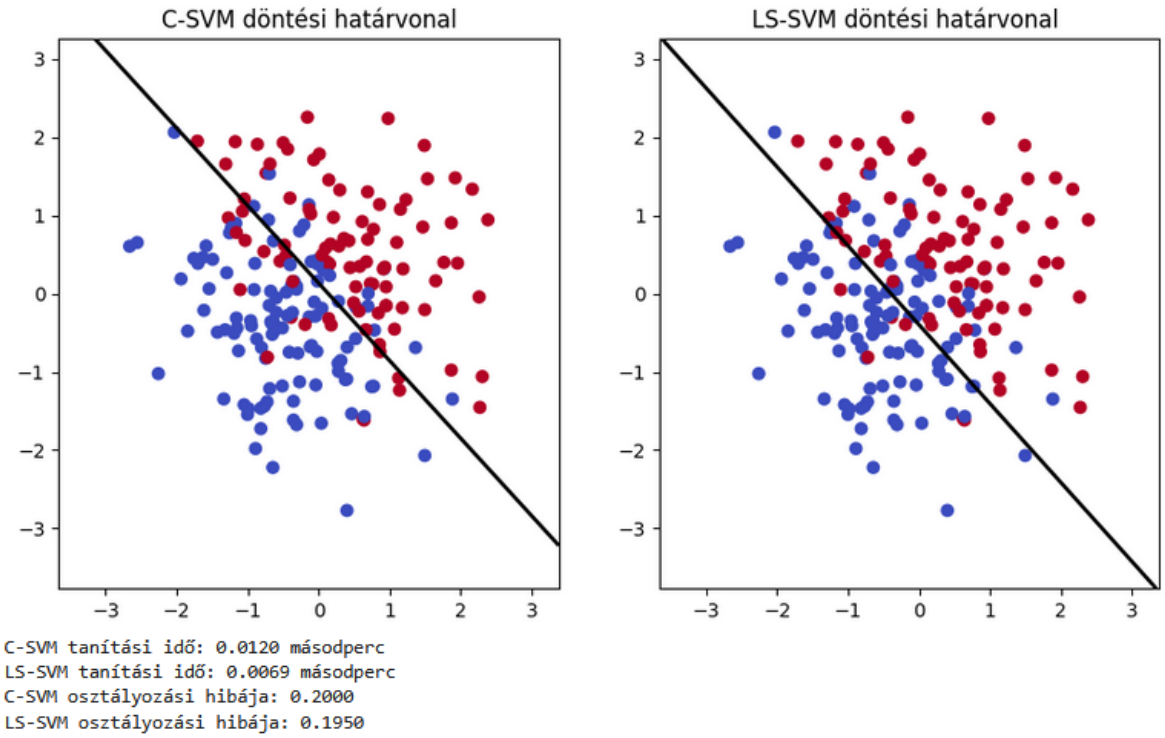
7.1. ábra. C-SVM és ν-SVM összehasonlítása

A 7.1 példából látható, hogy a szupport vektor arányra vonatkozó korlátozás mindig teljesül, azonban kis ν-kre a teszthiba bizonyos esetekben nagyobb lesz ν-nél. Ahogyan az is várható volt, nagyobb C esetén a teszthiba kissé mérséklődik. Amennyire hasonló a két típus, nem meglepő módon hasonló eredményeket többféle C, ν érték esetén.

7.2. C-SVM és LS-SVM

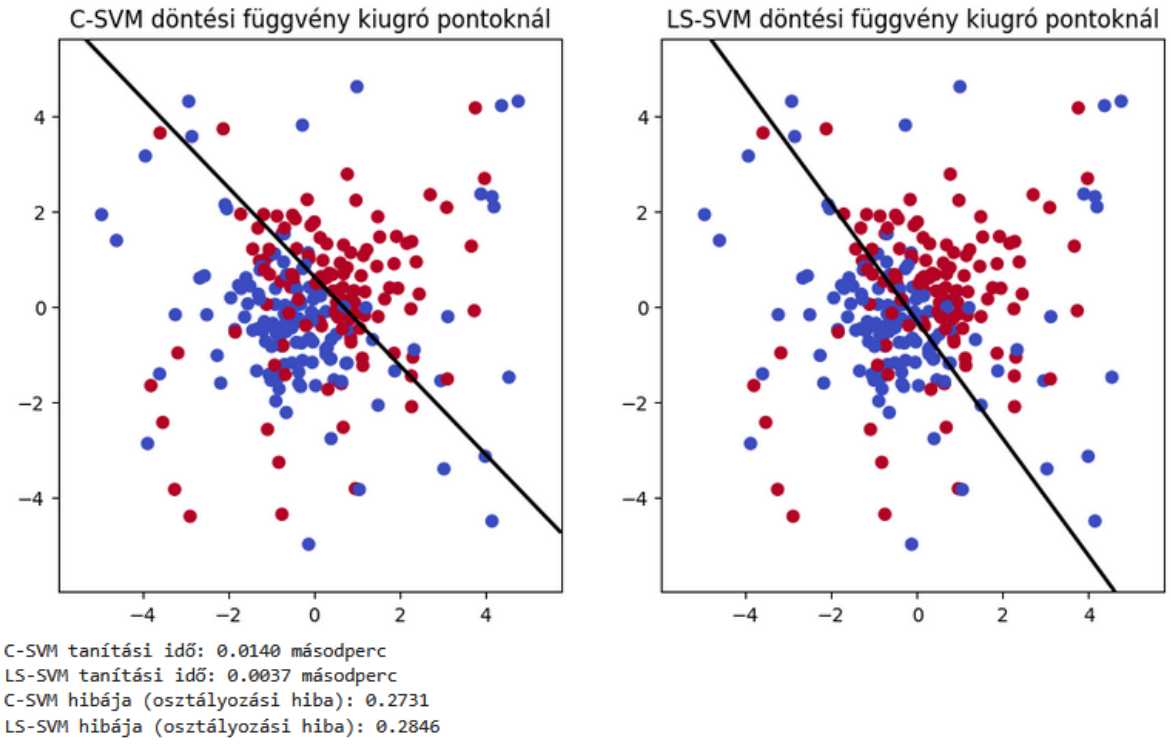
E feladatban alapvetően két dimenziós normál eloszlású az adatunk normál zajjal. A könnyebb ábrázolhatóság miatt a példában $n = 200$ adatponttal rendelkezünk. Egy kvadratikus programozási feladatot hasonlítunk össze lineáris egyenletek rendszerével. Az LS-SVM egyenlőségi feltételeket használ a C-SVM-től eltérően, amely egyenlőtlenségi feltételeket használ. Ezáltal LS-SVM-nél az összes adatpont szupport vektor, így ebben a példában felesleges a szupport vektorok hányadát mérni. LS-SVM esetén minden adatpont

hozzájárul a döntési függvényhez, ez rontja a skálázhatóságot. Láttuk, hogy az LS-SVM tanítása során mátrix invertálása szükséges, nagy n-ek esetében ez már kicsit problémásabb. E két faktort figyelembe véve, valamint, hogy az LS-SVM hajlamosabb túltanulásra, nagyobb n-ek esetében a C-SVM pontosabb. Egyszerűbb feltételek miatt az LS-SVM tanítása gyorsabb (ezt látjuk a 7.2-ben is). Azonban a példából is látható, nem feltétlen lesz a C-SVM mindig pontosabb. Látható az is, hogy mivel a két SVM különböző entitás, különböző döntési függvényekhez vezet még egyszerűbb példák esetében is, mégis eredményezhet ez hasonlóan jó teljesítményt.



7.2. ábra. C-sVM és LS-SVM döntés összehasonlítás

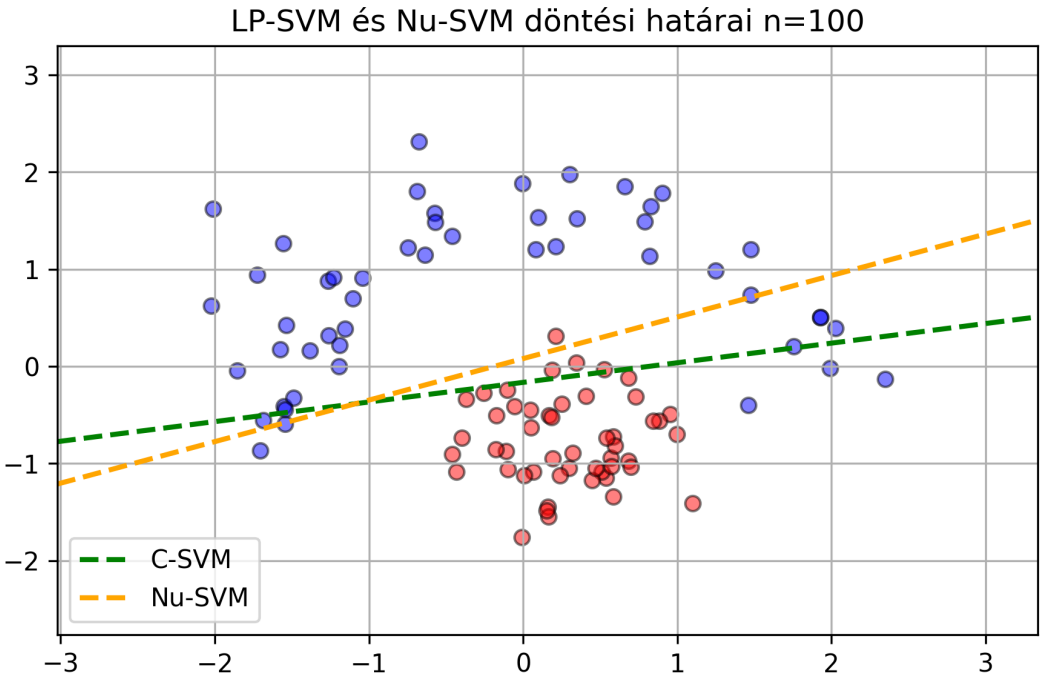
Mivel az LS-SVM négyzetes hibafüggvényt használ, ezzel szemben a C-SVM lineárisat, az LS-SVM kevésbé robosztus az outlier pontokkal szemben (kiugró pontok, amelyek jelentősen eltérnek a többi adatponttól). Így a nagy hibájú pontok erősebben befolyásolják az eredményt ebben az esetben. Visszatérve az előző adathalmazunkra, ha ennél a halmaznál a pontok legalább 20-30 százaléka kiugró, akkor az már megfordítja a versenyt a 2 SVM között. Ezt mutatja 7.3 is.



7.3. ábra. C-SVM és LS-SVM összevetés kiugró pontok esetében

7.3. ν-SVM és LP-SVM

A ν-SVM gyakorlatban sokkalta használtabb, mint az LP-SVM, ennek egyik fő oka a már említett intuitív paraméterbevezetés ν által. Azonban a lineáris programozás is egy széles körben kutatott és folyamatosan bővülő tudományterület, így bizonyos problémáknál érdemes az SVM-eket kiterjeszteni. AZ LP-SVM lineáris programozási feladatot old meg, így a futási ideje elég stabil még nagy n-ekre is, végig gyorsabb, mint a kvadratikus programozási feladatok. Az LP-SVM-ek nagy hasznot húznak a rapid LP-megoldókból.



LP-SVM

Mintaszám	Félreosztályozás hiba	Szupport vektorok aránya	Tanítási idő (s)
100	0.1000	0.4700	0.0008
500	0.1280	0.4860	0.0018
1000	0.1280	0.5410	0.0009
2000	0.1140	0.5125	0.0012
5000	0.1456	0.6174	0.0018

Nu-SVM (nu=0.5)

Mintaszám	Félreosztályozás hiba	Szupport vektorok aránya	Tanítási idő (s)
100	0.1000	0.4800	0.0007
500	0.1240	0.4960	0.0064
1000	0.1340	0.4940	0.0110
2000	0.1120	0.4910	0.0506
5000	0.1432	0.4848	0.3222

7.4. ábra. ν-SVM és LP-SVM összehasonlítás

Béta eloszlásból lett létrehozva a 2 átfedő osztály Laplace-zajjal tarkítva- ez az adatunk. Nem véletlen, hogy mivel az LS-SVM és a ν-SVM teljesen máshogyan közelíti meg a problémát, más döntési függvényt is kapunk. A 7.4 példán a futási időt vizsgálva látható, akárhányszorosára növeljük a mintaszámot az eredetihez képest, LP-SVM egyenletesen gyors, s meglehetősen pontos megoldást ad. Nagy n-ek esetében ahogy várható volt, a ν-SVM már kezd kicsit pontosabb lenni. ν-SVM esetén ν paramétert teszteltem és a legjobbnak tűnő ν érték lett kiválasztva összehasonlításra.

Összefoglalás

Szakdolgozatomban több témakör érintésével bevezettem a szupport vektor gépek működésének megértésehez szükséges fogalmakat, állításokat, majd elhoztam a különféle módosításait és össze is vetettem őket.

Az első fejezetben a statisztikai tanuláselméleti alapfogalmakkal foglalkoztam, mint veszteségfüggvény, kockázat, empirikus és regularizált kockázat. Láttuk, milyen fontos az, hogy milyen függvényosztály felett minimalizáljuk az empirikus kockázatot. Így jutottunk el VC-dimenzió, mint kapacitásfogalomig. A kockázatot empirikus kockázat és egy komplexitási tag segítségével becsültük. A hipersík osztályozókat vizsgálva, a strukturális kockázatminimalizálás kapcsán beláttuk ahhoz, hogy minimalizáljuk a komplexitási tagot, maximalizálnunk kell a margót. Ezzel megkaptuk a motivációt a maximális margó alapú lineáris osztályozókhoz, vagyis az SVM-ekhez. Ezután konvex optimalizálással foglalkoztunk, bevezetve a primál és duál feladat Lagrange-függvényes felírását, elégséges KKT-feltételeket. A harmadik fejezetben bevezettük a szupport vektor gépeket a kemény margó feladat kapcsán, amit a konvex optimalizálás eszközeivel megoldottunk eljutva a döntési függvény konkrét alakjához. Mivel ki akartuk terjeszteni az SVM-ek működését nemlineáris problémákra, jöttek a kernelek. A kerneleket a tulajdonságtérbe vetítés és skaláris szorzás összevonásaval vezettük be. A reprodukáló magú Hilbert-tér fontosabb tulajdonságait is listáztuk. Fontos tétel volt a reprezentációs tétel, miszerint a regularizált empirikus kockázatot minimalizáló függvény előáll a tanító adatokhoz tartozó reprodukáló kernelek lineáris kombinációjaként. Az SVM-ek kernellizált verziója is ezt alkalmazza. Ötödik fejezetben áttekintettük főbb változatait, mint C-SVM, ν-SVM, LS-SVM és LP-SVM. A következő fejezetben hoztunk több megközelítést, one-versus-one, one-versus-the rest és DAGSVM-et, amivel ki lehet terjeszteni a több osztályos osztályozási problémák megoldására az SVM-eket. Utolsó fejezetben megnéztünk pár gyakorlati feladatot, azokon keresztül hasonlítottuk össze feladatonként két eltérő SVM-típust.

Irodalomjegyzék

[1] Bernard Schölkolf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[2] Cristopher M. Bishop *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] Trevor Hastie, Robert Tibshirani and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[5] Shigeo Abe. *Support Vector Machines for Pattern Classification*. Chapter 4, Variants of Support Vector Machines. Springer, 2005.

[6] Csáji Balázs Csanád. *Data Mining and Machine Learning slides*. ELTE, 2024.

[7] Komornik Vilmos. *Valós analízis előadások I.* Typotex Kiadó, 2003.

[8] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

Alulírott Nagy Levente nyilatkozom, hogy szakdolgozatom elkészítése során az alább felsorolt feladatok elvégzésére a megadott MI alapú eszközöket alkalmaztam:

Feladat	Felhasznált eszköz	Felhasználás helye	Megjegyzés
Ábrakészítés	GPT o4-mini	6. fejezet	Megadatt parancs alapján egyszerű ábra készítése
Képminőség javítása	PixelBin	3. és 4. fejezet	Forrásokból kiszedett, helyenként kissé módosított ábráinak javítása

A felsoroltakon túl más MI alapú eszközt nem használtam.