CBS
COPENHAGEN
BUSINESS SCHOOL
HANDELSHØJSKOLEN

COPENHAGEN BUSINESS SCHOOL

KANDIDATAFHANDLING

CAND.MERC.(MAT.)

Support Vector Machine

- Teori & Implementering

Forfattere

Anders Hedegaard studie nr. 48404

Sebastian Kragh studie nr. 32716

Isabella Thuesen studie nr. 51233

Vejleder

Søren Feodor Nielsen

15. Maj 2018

Antal sider: 116

Abstract

Support Vector Machine is a popular method used for classification. In this paper, we develop an understanding of the theory behind the method of Support Vector Machine and its application. We find, that Support Vector Machine uses functions called kernels, to map data into a higher dimensional space in order to separate data with a hyperplane using linear classification theory. In order to construct a classification model, that can be applied well to other similar data, data is split into a training set, on which the model is built and a test set, on which the accuracy of the model is evaluated. Due to the properties of kernels, and the limited number of observations used for the construction of the model, Support Vector Machine is an efficient and robust method. To illustrate its application, we apply the Support Vector Machine classification to a set of financial data from the Orbis database, where we develop a model for classifying bankruptcy of Hotels. We compare the accuracy of different kernels and investigate the computational power required by Support Vector Machine.

Indhold

1 Motivation

Vores interesse for det teoretiske aspekt af kunstig intelligens herunder Machine Learning, skyldes kombinationen af den forholdsvis simple tilgang til implementeringen og de samtidig gode resultater sammenlignet med traditionelle statistiske metoder. Derudover finder vi det fascinerende, at man ved Machine Learning arbejder med algoritmer, der lærer og at Machine Learning er et område inden for statistikken, der er i rivende udvikling både med hensyn til teori og implementering. Vores særlige interesse er Support Vector Machine, herefter SVM, som er en af de mest populære metoder inden for Machine Learning.

Det praktiske aspekt af implementeringen interesserer os, fordi det muliggør en undersøgelse af, hvorvidt virksomheder inden for en særlig branche går konkurs, hvilket er et meget aktuelt emne i markedet. Emnet berører mange aktører og kan skabe indblik samt vigtig viden for både investorer, medarbejdere, ejere, kunder, myndigheder og samfundet som helhed. Med en klassifikationsmodel, der prædikterer en virksomheds konkurs præcist, bliver det muligt at tage forbehold for en potentiel konkurs eller at forsøge at afhjælpe denne.

Vi vil sige en særlig tak til vores vejleder Søren Feodor Nielsen for god og grundig vejledning. Søren Feodor Nielsen har motiveret os til at fokusere mere centralt på emnet, således at specialet kan betragtes som et mere teoretisk dybdegående speciale. Vejledningen muliggjorde, at vi kunne løse sværere problemstillinger undervejs.

2 Introduktion

Klassifikation handler om at finde forhold mellem observationers beskrivende variable og deres klasse i form af en klassifikationsmodel. Klassifikationen er især nyttig i de tilfælde, hvor man ønsker at bestemme observationers klasse. Udviklingen af klassifikationsmodellen kan ske ved forskellige metoder.

SVM er en metode, der bruges til klassifikation, hvor modellen bygges ved at lære ud fra observationer med klasser, der allerede er givet. Vi vil undersøge teorien bag SVM og dennes implementering i eksemplet for, hvorvidt et hotel er aktivt eller går konkurs.

I dette speciale vil vi først redegøre for data, herunder for den globale database vi har valgt, og efterfølgende de begrænsninger vi har lavet på data for at komme frem til en endelig specificering af data. Dernæst vil vi præsentere den forudgående teori, som vi bør kende til, før vi beskriver SVM. I den forudgående teori vil vi først forklare, hvad Machine Learning er og gennemgå vigtige definitioner herunder. Vi vil dernæst redegøre for vigtige resultater inden for optimeringsteori såsom betingelser for løsningen på kvadratiske optimeringsproblemer, der i SVM anvendes til at udvikle klassifikationsmodellen. Herefter vil vi forklare lineær klassifikationsteori, der anvendes i SVM, idet data klassificeres ved en adskillelse med et lineært hyperplan. Efterfølgende vil vi gennemgå den relevante teori om kerner, der bruges i SVM til at transformere data, således at data bliver nemmere at klassificere ved det lineære hyperplan. Her vil vi beskrive SVM med forskellige krav til dette adskillende hyperplan.

Vil også introducere teorien bag en algoritme, der implementerer SVM. Her vil vi omtale relevante emner i forhold til programmering af algoritmen, såsom den beregningskraft algoritmen kræver, samt hvordan de optimale værdier for forskellige parametre i klassifikationsmodellen vælges.

Vi vil dernæst illustrere teorien bag SVM ved at træne en klassifikationsmodel, der

på baggrund af et finansielt datasæt prædikterer konkurser for virksomheder. Til dette har vi valgt at benytte programmet R som software, da R er open-source og har pakker, hvor SVM allerede er implementeret.

Efterfølgende vil vi kort diskutere analysens resultater. Vi vil også diskutere metodens præcision, og de svagheder der opstår, ved anvendelsen af en SVM model som værktøj for forudsigelsen af konkurser.

3 Problemformulering

Formålet med dette speciale er at redegøre for teorien bag SVM på en overskuelig og forståelig måde. Dette vil vi gøre ved at fremlægge den nødvendige forudgående teori, som SVM gør brug af. Vi vil yderligere redegøre for implementeringen af SVM og illustrere metoden, ved at anvende denne til prædiktion af konkurser for virksomheder på baggrund af deres finansielle data. Til sidst vil vi diskutere nøjagtigheden af den model, vi kommer frem til, samt de udfordringer der er i forbindelse med implementeringen af SVM.

4 Metode

Dette speciale er hovedsagelig et teoretisk speciale. Vores fremgangsmåde er derfor, at gå i dybden med den forudgående teori for at kunne beskrive SVM. Den primære kilde, som vi bruger til teorien, er bogen

An Introduction to Support Vector Machines
and other kernel-based learning methods
Christianini, N. & Shawe-Taylor, J., Cambridge University Press 2000.

Vi vil derfor ikke foretage direkte kildereferencer til denne undervejs i specialet.

For at kunne sige noget kvalificeret om fordele og ulemper ved SVM sammenlignet med andre statistiske metoder, er det nødvendigt at udvikle sådanne modeller med omhu. Da vi har valgt at begrænse os til at fokusere på SVM som metode, har vi derfor ikke lavet en sammenligning af SVM mod andre statistiske metoder.

I specialet vil vi til sidst forsøge at analysere implementeringen af SVM på et datasæt. Formålet er dog ikke at foretage en udtømmende analyse af vores datasæt. Analysen har derimod som mål at illustrere den teori, der ligger bag ved SVM.

5 Data

I dette afsnit vil vi introducere data og dets begrænsninger, for at kunne specificere et udtræk, som vi senere vil bruge i specialet.

Når man arbejder med SVM, er det vigtigt at have data af passende kvalitet og mængde. Forberedelse af data er derfor ofte nødvendigt. Hvis vi ikke forbereder vores data, kan vi risikere, at data vil være af lav kvalitet med manglende observationer. Dette kan have indflydelse på præcisionen af vores analyse eller kan ligefrem føre til forkerte konklusioner.

Vi benytter den samme database til udtræk af al vores data, hvilket vil gøre, at processen ikke tager så lang tid, som hvis vi skulle have vores data fra flere forskellige databaser.

5.1 Database

Da vi ønsker at prædiktere konkurser ved SVM, er det vigtigt at finde data, der opfylder de krav, der gør det muligt at lave denne form for analyse. Vi skal have data, der repræsenterer hele populationen, og data der gør det muligt at se status, hvor status indikerer, hvorvidt virksomheden er aktiv eller konkurs. Herudover skal der være nok data til at muliggøre en statistisk analyse.

En database, der opfylder disse krav, er Orbis fra Bureau van Dijk. Orbis er en global database, som indeholder data omkring mange millioner af virksomheder fra forskellige brancher. Databasens størrelse og det, at den indeholder data på virksomheders status, gør den ideel til formålet.

5.2 Begrænsninger

Som en del af forberedelsen af data er vi nødt til at begrænse vores data, fordi det kan være svært at sammenligne virksomheder fra forskellige brancher og/eller forskellige lande. Vi opsætter derfor en række kriterier, der begrænser vores data på baggrund af disse overvejelser.

Status

Først og fremmest må vi definere, hvornår en virksomhed er konkurs. Databasen Orbis tilbyder mange forskellige valg for både aktive og inaktive virksomheder. Eftersom vi er interesseret i konkurser, vil vi kun overveje de virksomheder, der udtrykkeligt er defineret som konkurs. Det gør vi i håb om at undgå specielle tilfælde, der kunne generere outliere, dvs. virksomheder der skiller sig særligt ud fra de andre.

Hvad angår virksomheder, der ikke er konkurs, vil vi igen kun overveje virksomheder, der udtrykkeligt er defineret som aktive. Dette sikrer os, at vi beskæftiger os med de mere sunde virksomheder, og det vil derfor hjælpe med at klassificere dem mod de konkursramte virksomheder.

Definition på konkurs

En virksomhed kan erklæres konkurs, når denne er *insolvent*. En virksomhed er insolvent, når den i en længere periode ikke har kunnet betale sine kreditorer.

Branche

Vi vil undersøge en branche, der har en stor andel homogene virksomheder, og som samtidigt indeholder en stor mængde data. Hotelbranchen er en branche, der opfylder begge krav, og som samtidig er et udmærket investeringsmål for investorer.

Lokation

Som før nævnt er Orbis en global database, hvilket betyder, at denne indeholder virksomheder fra hele verden. Da der er store forskelle på virksomheder fra eksempelvis Norge og Zimbabwe, begrænser vi os ved at vælge ét land - Frankrig. Vi vælger

Frankrig, primært fordi mængden af data for hoteller er tilfredsstillende.

Periode

For at sikre os temporal robusthed overfor økonomiske ændringer trækker vi data for perioden fra den 01/01/2000 - 01/01/2018. Når vi tager en så lang periode, er vi ikke så udsat overfor ændringer i konjunkturen på samme måde, som hvis vi havde data fra en periode, hvor markedet var i enten lav- eller højkonjunktur.

5.2.1 Specificering af Data

For at kunne foretage klassifikation ved SVM er vi nødt til at have en række forklarende variable, hvormed vi kan lave vores analyse. Dataudtrækket, som vi bruger i specialet, er trukket den 17/04/2018 og indeholder variable, der eksempelvis angiver hvert hotels profit og likviditet. Derudover har vi en bred vifte af andre forklarende variable med for at kunne beskrive den enkelte virksomhed bedst muligt.

Vi fjerner alle de observationer, der mangler data på de valgte variable. Vi fjerner derfor de variable, hvor der enten har været for meget få eller ingen observationer først. Det sikrer os, at vi kan beholde et passende antal observationer i datasættet. Ved først at fjerne variable sikrer vi os ikke bare et mere komplet datasæt, vi mindsker også sandsynligheden for bias, da der vil være større sandsynlighed for at fjerne en bestemt type data. Det kunne eksempelvis være, at der var en variabel, hvor det kun var de største hoteller, der havde data. På den måde ville vi fjerne alle de mindre hoteller og derved skabe bias i datasættet.

Herefter fjerner vi de variable, som åbenlyst afhænger af hinanden. Et eksempel er variablen *pengestrøm over driftsindtægter*, der perfekt beskrives af variablene *pengestrøm* og *driftsindtægter*. Fjerner vi ikke denne variabel, har vi et datasæt med perfekt multikollinearitet, hvilket er kendt for at kunne forårsage variansinflation. SVM er ikke påvirket af kollinearitet i samme grad som andre statistiske modeller,

men det anbefales dog at undgå dette, især når det gælder perfekt multikollinearitet.

Når alle disse overvejelser er gjort, ender vi med et datasæt med 18 forklarende variable og 1 responsvariabel. Heri har vi 1.012 observationer, hvoraf 506 har status konkurs, og 506 har status aktiv. Vi har valgt 506 variable af hver status af to grunde. Den første grund er, at efter vi har fjernet observationer med manglende data, er der 506 konkurs observationer tilbage. Den anden grund er, at vi vælger en stikprøve på 506 aktive observationer. Dette gør vi, fordi vi ønsker et balanceret datasæt, da et datasæt der er ubalanceret, som regel vil have unaturligt høje klassifikationsrater. Dette datasæt benyttes i afsnit 9, og de forklarende variable kan findes i bilag A, afsnit 13.1.

6 Forudgående teori

For at kunne forklare SVM som metode er det nødvendigt først at redegøre for den bagvedliggende teori. Vi vil i dette kapitel redegøre for denne forudgående teori.

6.1 Machine Learning

Helt overordnet handler Machine Learning om at anvende algoritmer, der finder mønstre i data ud fra det input, algoritmerne fodres med. Disse algoritmer betegnes learning machines. Populært sagt lærer learning machines fra eksempler.

Til forskel fra traditionelle statistiske modeller vil man med Machine Learning oftest ende med en matematisk modellering af problemet, hvor sammenhængen mellem de forklarende variable og responsvariablen vil være uigennemskuelig. Machine Learning bruges derfor ofte ved problemer, hvor fortolkning af de forklarende variables indflydelse på responsvariablen ikke er interessant. Dette er eksempelvis tilfældet ved Facebooks ansigtsgenkendelse, hvor Machine Learning bruges til at klassificere billeder i klasserne ansigt og ikke ansigt. Her er man ikke interesseret i, hvilken indflydelse farven på forskellige pixels og disses placering har på, om der er tale om et ansigt eller ej.

SVM hører under den del af Machine Learning, der hedder *Supervised Learning*. I Supervised Learning bruges observationers forklarende variable til at tildele observationerne en af de prædefinerede klasser - ansigt eller ikke ansigt i Facebook eksemplet. I eksemplet betyder dette, at klassifikationsmodellen er konstrueret ud fra billeder, hvor ansigter allerede er udpeget af mennesker. Dette adskiller Supervised Learning fra *Unsupervised Learning*, hvor der ikke findes prædefinerede klasser. Overordnet set handler Unsupervised Learning om learning machines, der selv skal finde klasser - eller mere generelt mønstre i data, hvor Supervised Learning bliver "vejledt" med et prædefineret output.

Hvor målet for Unsupervised Learning er at finde mønstre i data, der allerede er til rådighed, er målet med Supervised Learning at prædiktere nye observationers klasse. Gives modellen en række nye billeder uploadet af Facebook-brugere, skal modellen altså kunne afgøre, hvad der er ansigt, og hvad der ikke er.

SVM er en metode under Machine Learning, der træner en klassifikationsmodel ud fra et sæt observationer. Klassifikationsmodellen består af et sæt *supportvektorer*, der sammen danner et n-dimensionalt hyperplan, som adskiller observationerne. Klassifikationsmodellen måles nu både på, hvor godt denne klassificerer de kendte observationer, men også på hvor godt modellen kan *generaliseres* til klassificering af nye observationer.

6.1.1 Begreber

Datasættet, som klassifikationsmodellen konstrueres ud fra, kaldes *træningssættet*, og processen kaldes *træning*. Når modellen er trænet, testes den på nye ukendte observationer. Sættet af de nye observationer kaldes *testsættet*. Vi betegner træningssættet S og testsættet T.

Målet for træningen er at udvikle en model, der klassificerer observationerne korrekt for både træningssættet og testsættet. Raten, hvormed observationerne klassificeres korrekt, kaldes *klassifikationsraten* og er givet ved antallet af korrekt klassificerede observationer divideret med det samlede antal observationer i procent. Gode klassifikationsmodeller er modeller med høj klassifikationsrate på trænings- og testsættet.

Resultatet af træningen, nemlig klassifikationsmodellen, består af en funktion kaldet *beslutningsfunktionen*, der ud fra den specifikke observations forklarende variable tildeler observationen en klasse. *Beslutningsreglen* er defineret som fortegnet på værdien af beslutningsfunktionen, der vil afgøre observationens klasse.

6.2 Optimeringsteori

Det vil vise sig, at problemet, som vi skal løse i SVM, er et kvadratisk optimeringsproblem med lineære bibetingelser. Vi vil derfor i dette afsnit opstille betingelser for, at der ikke findes lokale minima, og at løsningen dermed vil være optimal. Vi skal desuden se på den duale repræsentation af optimeringsproblemet, som viser sig at være nyttig i forhold til løsningen af optimeringsproblemet i SVM.

Vi starter med at definere en affin funktion og en konveks mængde.

6.2.1 Definitioner

Affin funktion

En funktion er *affin*, når den kan skrives på følgende form

$$f(\overrightarrow{w}) = \mathbf{A}\overrightarrow{w} + \overrightarrow{b},$$

hvor $\mathbf{A}$ er en matrix, og $\overrightarrow{b}$ og $\overrightarrow{w}$ er vektorer. Affine funktioner vil altid være konvekse (og konkave), da deres Hessematrix er 0.

Konveks mængde

En mængde $\Omega \subseteq \mathbb{R}$ er konveks, når det opfyldes, at $\forall\ \overrightarrow{w}, \overrightarrow{u} \in \Omega$ og $\theta \in (0, 1)$,

$$\theta\overrightarrow{w} + (1 - \theta)\overrightarrow{u} \in \Omega.$$

6.2.2 Kvadratisk optimering

For at løse et optimeringsproblem med en kvadratisk objektfunktion og lineære bibetingelser kan vi benytte Lagrange og Karush-Kuhn-Tucker betingelserne.

Mens Lagrange gør det muligt at beskæftige sig med kvadratiske objektfunktioner, så muliggøre Karush-Kuhn-Tucker betingelserne at operere med bibetingelser, der har ulighedstegn.

Lagrangefunktionen

Lagrangefunktionen kan opstilles ud fra optimeringsproblemet. Det er nemlig behjælpeligt at konvertere det betingede optimeringsproblem til en Lagrangefunktion, der ikke vil have bibetingelser, for at løse problemet. Optimeringsproblemet skrives generelt som

$$\begin{aligned} \min \quad & f(\overrightarrow{w}), && \overrightarrow{w} \in \Omega \\ \text{ubb.} \quad & g_i(\overrightarrow{w}) \leq 0, && i = 1, ..., k \\ & h_i(\overrightarrow{w}) = 0, && i = 1, ..., m. \end{aligned}$$

Lagrangefunktionen bliver nu

$$L(\overrightarrow{w}, \overrightarrow{\alpha}, \overrightarrow{\beta}) = f(\overrightarrow{w}) + \sum_{i=1}^{k} \alpha_i g_i(\overrightarrow{w}) + \sum_{i=1}^{m} \beta_i h_i(\overrightarrow{w}), \qquad \alpha_i, \beta_i \in \mathbb{R}.$$

Vi vil kalde denne opstilling for den *primale repræsentation* af optimeringsproblemet eller blot det *primale problem*, som står i modsætning til den *duale repræsentation*, som vi på samme måde vil referere til som det *duale problem*. Vi kalder $\overrightarrow{w}$ for den *primale variabel*. Det er dog muligt at have flere primale variable, hvor vi i SVM vil have mere end en.

6.2.3 Den duale repræsentation

Vi vil nu vise, at vi kan omskrive det primale problem, således at $\overrightarrow{w}$ ikke længere vil indgå i problemet. Omskrivningen kaldes den *duale repræsentation* af optimeringsproblemet. Dette vil vise sig nyttigt, da det simplificerer optimeringsproblemet.

Fremgangsmåden er at differentiere Lagrangefunktionen med hensyn til de primale variable, i dette tilfælde $\overrightarrow{w}$, for derefter at sætte differentialerne lig nul. Således

kan vi udtrykke $\overrightarrow{w}$ ved Lagrangemultiplikatorerne $\overrightarrow{\alpha}$ og $\overrightarrow{\beta}$, som vi kalder de *duale variable*. Dermed kan vi skrive den primale variabel ud af optimeringsproblemet, og i stedet løse problemet for de duale variable i det duale problem defineret ved

$$\begin{aligned} \max \quad & \theta(\overrightarrow{\alpha}, \overrightarrow{\beta}) \\ \text{ubb.} \quad & \alpha_i \geq 0, \quad i = 1, ..., k \end{aligned}$$

med $\theta(\overrightarrow{\alpha}, \overrightarrow{\beta})$ defineret ved

$$\theta(\overrightarrow{\alpha}, \overrightarrow{\beta}) = \inf_{\overrightarrow{w} \in \Omega} L(\overrightarrow{w}, \overrightarrow{\alpha}, \overrightarrow{\beta}).$$

Vi vil nu bevise, at hvis $\overrightarrow{\alpha}$ og $\overrightarrow{\beta}$ er mulige løsninger til det duale problem, samt at $\overrightarrow{w}$ er en mulig løsning til det primale optimeringsproblem, så vil det altid gælde, at $f(\overrightarrow{w}) \geq \theta(\overrightarrow{\alpha}, \overrightarrow{\beta})$.

Bevis

Givet vektorerne $\overrightarrow{w}$, $\overrightarrow{u}$, $\overrightarrow{\alpha}$ og $\overrightarrow{\beta}$, hvor $\overrightarrow{w}$ er en mulig løsning til det primale optimeringsproblem og $\overrightarrow{u}, \overrightarrow{w} \in \Omega$, mens $\overrightarrow{\alpha}$ og $\overrightarrow{\beta}$ er løsninger til det duale optimeringsproblem, kan vi ved definitionen på infimum skrive

$$\begin{aligned} \theta(\overrightarrow{\alpha}, \overrightarrow{\beta}) &= \inf_{\overrightarrow{u} \in \Omega} L(\overrightarrow{u}, \overrightarrow{\alpha}, \overrightarrow{\beta}) \\ &\leq L(\overrightarrow{w}, \overrightarrow{\alpha}, \overrightarrow{\beta}) \\ &= f(\overrightarrow{w}) + \sum_{i=1}^{k} \alpha_i g_i(\overrightarrow{w}) + \sum_{i=1}^{m} \beta_i h_i(\overrightarrow{w}) \\ &\leq f(\overrightarrow{w}), \end{aligned}$$

hvor sidste ulighed kommer af, at $\overrightarrow{\alpha} \geq 0$, og at bibetingelserne per definition er $g(\overrightarrow{w}) \leq 0$ og $h(\overrightarrow{w}) = 0$, med $\overrightarrow{w}$ defineret til at opfylde disse bibetingelse, da denne er en mulig løsning. Hvis løsningen på det duale og primale problem er ens, dvs. $f(\overrightarrow{w}) = \theta(\overrightarrow{a}, \overrightarrow{\beta})$, bliver den sidste ulighed en lighed

$$f(\overrightarrow{w}) + \sum_{i=1}^{k} \alpha_i g_i(\overrightarrow{w}) + \sum_{i=1}^{m} \beta_i h_i(\overrightarrow{w}) = f(\overrightarrow{w}),$$

og således har vi opnået optimum. Dette holder dog kun hvis

$$\forall i = 1, ..., k : \ \alpha_i^* g_i(\overrightarrow{w}^*) = 0.$$

■

Svag dualitet

Når $\sum_{i=1}^{k} \alpha_i g_i(\overrightarrow{w}) < 0$ betyder det i stedet, at løsningen til det duale problem ikke er lig løsningen til det primale problem, dvs.

$$\theta(\overrightarrow{\alpha}^*, \overrightarrow{\beta}^*) < f(\overrightarrow{w}^*)$$
$$\Rightarrow f(\overrightarrow{w}^*) - \theta(\overrightarrow{\alpha}^*, \overrightarrow{\beta}^*) > 0.$$

Derfor siger vi, at hvis $\sum_{i=1}^{k} \alpha_i g_i(\overrightarrow{w}) < 0$, så er dualiteten af problemet svag, da der vil være forskel mellem løsningen til det duale problem og det primale problem.

For at vi kan bruge løsningen på det duale problem, skal vi sikre os, at $\theta^* = f^*$. Den stærke dualitetssætning giver betingelserne for, hvornår dette er tilfældet.

Den stærke dualitetssætning

For et optimeringsproblem defineret på en konvekst mængde $\Omega \subseteq \mathbb{R}$

$$\begin{array}{lll} \min & f(\overrightarrow{w}), & \overrightarrow{w} \in \Omega \\ \text{ubb.} & g_i(\overrightarrow{w}) \leq 0, & i = 1, ..., k \\ & h_i(\overrightarrow{w}) = 0, & i = 1, ..., m, \end{array}$$

hvor g_i og h_i er affine funktioner, så er $\theta^* = f^*$.

Af disse resultater fremkommer Karush-Kuhn-Tucker betingelserne, der giver betingelserne for en optimal løsning til det kvadratiske optimeringsproblem.

6.2.4 Karush-Kuhn-Tucker betingelserne

Givet et optimeringsproblem, defineret på en konveks mængde $\Omega \subseteq \mathbb{R}$,

$$\begin{aligned} \min \quad & f(\overrightarrow{w}), && \overrightarrow{w} \in \Omega \\ \text{ubb.} \quad & g_i(\overrightarrow{w}) \leq 0, && i = 1, ..., k \\ & h_i(\overrightarrow{w}) = 0, && i = 1, ..., m, \end{aligned}$$

med f konveks og g_i og h_i affine, er de nødvendige og tilstrækkelige betingelser for at $\overrightarrow{w}^*$ er et optimum, at der eksisterer et $\overrightarrow{\alpha}^*$ og et $\overrightarrow{\beta}^*$, der opfylder, at den differentierede Lagrangefunktion med hensyn til $\overrightarrow{w}$ og $\overrightarrow{\beta}$ giver

$$\frac{\partial L(\overrightarrow{w}^*, \overrightarrow{\alpha}^*, \overrightarrow{\beta}^*)}{\partial \overrightarrow{w}} = 0 \tag{1}$$

$$\frac{\partial L(\overrightarrow{w}^*, \overrightarrow{\alpha}^*, \overrightarrow{\beta}^*)}{\partial \overrightarrow{\beta}} = 0, \tag{2}$$

og at

$$\alpha_i^* g_i(\overrightarrow{w}^*) = 0, \quad i = 1, ..., k \tag{3}$$

$$g_i(\overrightarrow{w}^*) \leq 0, \quad i = 1, ..., k \tag{4}$$

$$\alpha_i^* \geq 0, \quad i = 1, ..., k. \tag{5}$$

Aktive og inaktive bibetingelser

En bibetingelse $g_i(\overrightarrow{w}) \leq 0$ i et optimeringsproblem siges at være *aktiv*, hvis løsningen opfylder bibetingelsen med strengt lighedstegn, dvs. $g_i(\overrightarrow{w}) = 0$, og *inaktiv* hvis $g_i(\overrightarrow{w}) < 0$.

Karush-Kuhn-Tucker komplementaritetsbetingelsen

Ligning (3) kaldes Karush-Kuhn-Tucker komplementaritetsbetingelse. Denne siger, at når vores bibetingelse $g_i(\overrightarrow{w}) \leq 0$ er aktiv, dvs. $g_i(\overrightarrow{w}) = 0$, så kan α_i^* tage værdierne $\alpha_i^* \geq 0$. Er bibetingelsen i stedet inaktiv, medfører det, at $\alpha_i^* = 0$.

Da Lagrangemultiplikatorerne $\overrightarrow{\alpha}$ og $\overrightarrow{\beta}$ blot ganges på bibetingelserne, angiver disse løsningens sensitivitet i forhold til en ændring i den pågældende bibetingelse. α_i^*

angiver dermed den optimale løsnings sensitivitet over for ændringer i den tilhørende bibetingelse g_i. Ændres en bibetingelse med et $\alpha_i = 0$ har det dermed ingen betydning for den optimale løsning.

I SVM skal vi senere se, at hver bibetingelse kommer af observationer, og at løsningen bliver et hyperplan, der separerer og dermed klassificerer observationer. Dette betyder, at det kun er de observationer med $\alpha_i > 0$, der bestemmer det separerende hyperplan. Disse indflydelsesrige observationer betegnes som *supportvektorer*.

Når observationerne klassificeres i SVM, foregår dette ved en lineær klassifikation af observationerne eller en transformation af disse. Vi vil starte med at redegøre for lineær klassifikation og dermed vente med at beskrive, hvordan data transformeres under SVM.

6.3 Lineær Klassifikation

Lineær klassifikation er en metode, hvorpå vi kan klassificere vores data i to klasser, således at vi kan adskille data med et separerende hyperplan.

Det antages, at data X stammer fra et rum med alle reelle tal, dvs. $X \subseteq \mathbb{R}^n$. Den del af data, der er udtaget til at være træningssæt, skrives som $S = (\overrightarrow{x}_1, ..., \overrightarrow{x}_n)$, hvor $S \in X$, og hvor hver vektor $\overrightarrow{x}_i = (x_1, .., x_m)$ definerer en observation.

Ved lineær klassifikation vil man tildele hver observation en klasse. Observationens klasse betegnes ved $y_i \in Y$, hvor

$$Y = \{-1, 1\},$$

hvilket vil sige, at y_i angiver, om observation i er i den positive eller negative klasse. Når hver observation er tildelt en klasse, kan vi opskrive træningssættet på følgende måde

$$S = ((\overrightarrow{x}_1, y_1), ..., (\overrightarrow{x}_n, y_n)) \subseteq (X \times Y)^n,$$

hvor n er antallet af observationer.

For at tildele en specifik observation $\overrightarrow{x}$ en klasse bruges den lineære beslutningsfunktion $f : X \subseteq \mathbb{R}^n$. Hvis $f(\overrightarrow{x}) \geq 0$, så er $\overrightarrow{x}$ blevet klassificeret til at tilhøre den positive klasse, og hvis $f(\overrightarrow{x}) < 0$, så er $\overrightarrow{x}$ klassificeret til at tilhøre den negative klasse.

Den lineære funktion $f(\overrightarrow{x})$ kalder vi beslutningsfunktionen, og med $\overrightarrow{x} \in X$ skriver vi beslutningsfunktionen som

$$\begin{aligned} f(\overrightarrow{x}) &= \langle \overrightarrow{w} \cdot \overrightarrow{x} \rangle + b \\ &= \sum_{i=1}^{m} w_i x_i + b, \end{aligned} \tag{6}$$

hvor parametrene i funktionen, $\overrightarrow{w}$ og b, er vektorer, og hvor $\text{sgn}(f(\overrightarrow{x}))$ er vores beslutningsregel. Beslutningsreglen er defineret ved

$$\text{sgn}(f(x)) = \begin{cases} +1, & f(x) \geq 0 \\ -1, & f(x) < 0 \end{cases}.$$

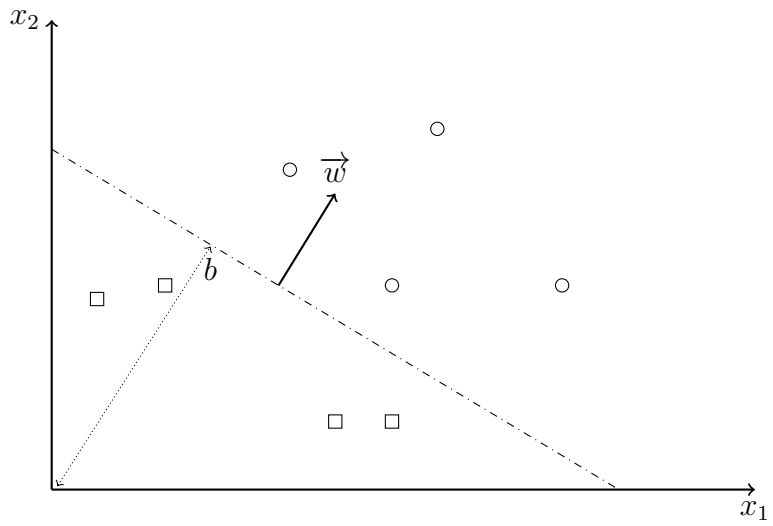
Vi introducerer herunder de lineære hyperplaner, der lineært skal separere observationerne i klasserne.

6.3.1 Separerende lineære hyperplaner

Det lineære separerende hyperplan er et hyperplan, der separerer den positive og negative klasse i rummet. Et hyperplan af enhver dimension kan skrives som

$$\langle \overrightarrow{w} \cdot \overrightarrow{x} \rangle + b = 0,$$

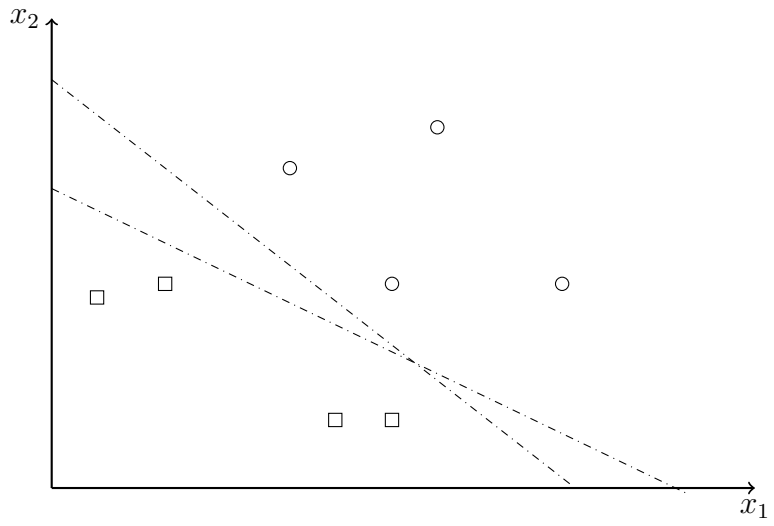
hvor vægten $\overrightarrow{w}$ er en vektor, der ligger ortogonalt på hyperplanet, og som styrer hældningen på hyperplanet, mens b forskyder hyperplanet parallelt til sig selv.



Figur 1

Figur 1 herover illustrerer et todimensionalt rum, med observationer der er tildelt en klasse. Observationerne markeret med firkanter repræsenterer den ene klasse, og observationerne markeret med cirkler repræsenterer den anden klasse.

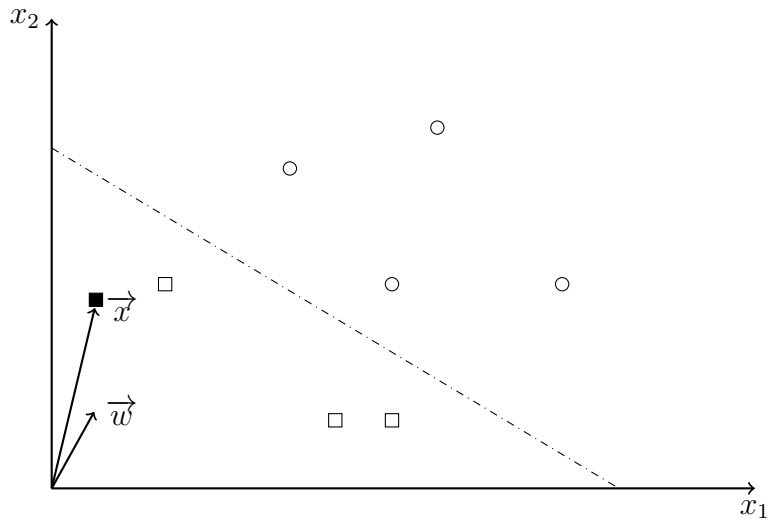
Klasserne separeres af et lineært hyperplan, der er illustreret ved den stiplede linje. Hældningen og placeringen af det lineære hyperplan kan variere, og flere forskellige hyperplaner vil kunne separere klasserne. Figur 2 herunder viser to andre lineære hyperplaner, der ligeledes separerer klasserne fra figur 1.



Figur 2

Data, der kan adskilles lineært, har således uendeligt mange hyperplaner, der separerer data.

Som nævnt i starten af afsnittet tildeles hver observation enten den positive eller negative klasse. Hvorvidt observationer ligger på den ene eller anden side af det separerende hyperplan angiver, hvilken klasse observationen tilhører.



Figur 3

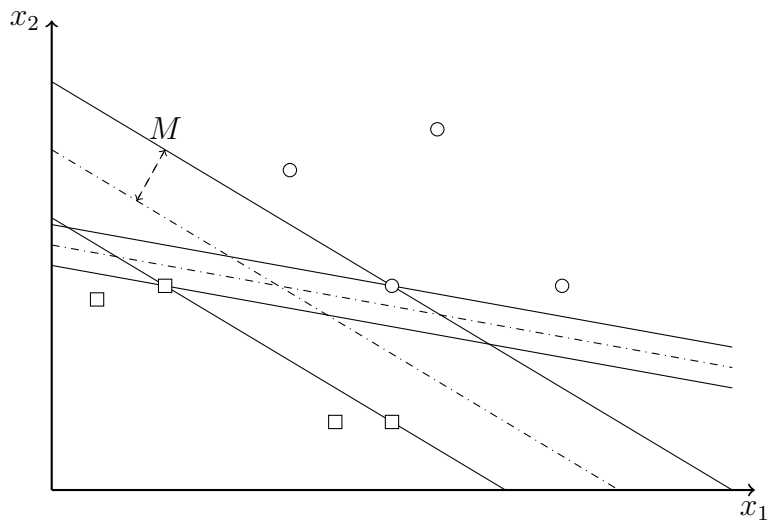
Kigger vi på figur 3, har vi en vilkårlig observation $\overrightarrow{x}$. Vi kan nu udregne, hvorvidt den vilkårlige observation ligger på den ene eller den anden side af hyperplanet. Dette gøres ved at udregne, hvorvidt værdien af $\langle \overrightarrow{w} \cdot \overrightarrow{x} \rangle + b$ er positiv eller negativ.

Når $\langle \overrightarrow{w} \cdot \overrightarrow{x} \rangle + b$ er større end eller lig nul, har vi en observation, der tilhører den positive klasse og ligger på den ene side af det separerende hyperplan. Hvis $\langle \overrightarrow{w} \cdot \overrightarrow{x} \rangle + b$ er mindre end nul, har vi en observation, der tilhører den negative klasse og ligger på den anden side af det separerende hyperplan. Som før nævnt er $\overrightarrow{x}$ en vilkårlig observation, og $\overrightarrow{x}$ er kendt, mens længden af $\overrightarrow{w}$ og b er ubekendte.

Vi bruger nu notationen $\overrightarrow{x}_+$, når den vilkårlige observations klasse er positiv og ligger på den ene side af det separerende hyperplan, og $\overrightarrow{x}_-$ når observationens klasse er negativ og dermed ligger på den anden side af det separerende hyperplan.

6.3.2 Den maksimale geometriske margin

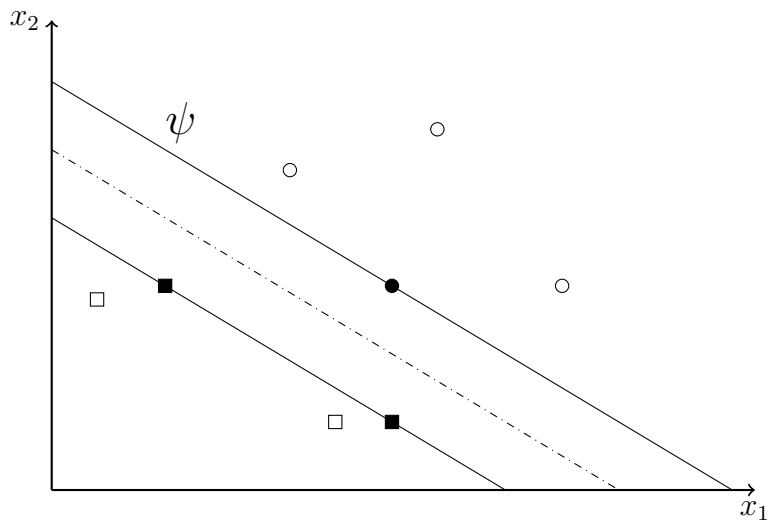
Der er flere metoder, hvormed man kan finde frem til det bedst klassificerende hyperplan. SVM definerer det bedste hyperplan, som det hyperplan hvor *marginen* er størst. Marginen består af to hyperplaner, som er parallelt forskudt ud til den eller de observationer, der ligger tættest på det separerende hyperplan på henholdsvis den ene og den anden side. Hyperplanet vil ligge midt imellem marginen.



Figur 4

På figuren ovenfor ses to forslag til separerende hyperplaner vist med stiplede linjer, og hvor marginerne er illustreret ved fuldt optrukne linjer. De to separerende hyperplaner har forskellig afstand ud til den tilhørende margin. Denne afstand kaldes den *geometriske margin*, M. En metode, hvormed vi kan finde det bedst separerende hyperplan, er altså at maksimere den geometriske margin.

I figur 5 markerer vi observationerne, der ligger på marginen, med sort udfyldning. Disse observationer er dem, der bestemmer hældning og placering af hyperplanet, da de ligger tættest på hyperplanet. Dette er en vigtig pointe, da det betyder, at det kun er nogle af observationerne, der skal bruges for at bestemme hyperplanet. Disse observationer kaldes *supportvektorer*.



Figur 5

Vi kalder funktionen for marginen for den *funktionelle margin*, og definerer denne ved

$$\psi = y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b), \qquad \psi \in \mathbb{R}, i = 1, ..., n.$$

Da vi benytter tilgangen, hvor marginen skal maksimeres, svarer det til at maksimere afstanden mellem marginen og det separerende hyperplan, dvs. den geometriske margin M. Af definitionen på marginen, må der ikke ligge punkter imellem marginen og hyperplanet. Dette giver maksimeringsproblemet

$$\begin{aligned} \max \quad & M \\ \text{ubb.} \quad & y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) \geq M, \qquad i = 1, ..., n. \end{aligned}$$

Uden at ændre på problemet, er det tilladt at skalere $(\overrightarrow{w}, b)$ til $(a \cdot \overrightarrow{w}, a \cdot b)$ eftersom

$$\text{sgn}(\overrightarrow{w} \cdot \overrightarrow{x} + b) = \text{sgn}(a \cdot \overrightarrow{w} \cdot \overrightarrow{x} + a \cdot b), \quad \forall a \in \mathbb{R}^+.$$

Vi kan derfor fastsætte den funktionelle margin til at være 1, hvilket medfører

$$y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) = 1, \tag{7}$$

som vi vil benytte i ligning (8) og (9).

I afsnittet om separerende lineære hyperplaner så vi, at en observation, der var klassificeret i enten den positive eller negative klasse, kunne skrives som $\overrightarrow{x}_+$ eller $\overrightarrow{x}_-$, alt efter hvilken side af hyperplanet observationen var placeret på.

Af ligning (7) ses det nu, at en observation, der ligger på marginen, opfylder

$$\langle \overrightarrow{w} \cdot \overrightarrow{x}_+ \rangle + b = 1,$$

og

$$\langle \overrightarrow{w} \cdot \overrightarrow{x}_- \rangle + b = -1,$$

da $y_i = \{-1, 1\}$.

Der vil ikke ligge observationer mellem marginen og hyperplanet, da marginen er bestemt af observationerne, der ligger nærmest hyperplanet. Dette medfører ligningerne

$$\langle \overrightarrow{w} \cdot \overrightarrow{x}_+ \rangle + b \geq 1 \tag{8}$$

$$\langle \overrightarrow{w} \cdot \overrightarrow{x}_- \rangle + b \leq -1 \tag{9}$$

Den binære variabel $y_i = \{-1, 1\}$, hvor $i = 1, ..., n$, som vi fra lineær klassifikation ved, betegner observationernes klasse, vil tage værdien 1, hvis observationen er i den positive klasse og vil tage værdien -1, hvis observationen er i den negative klasse. Denne information leder os videre til følgende funktioner

$$\begin{aligned} (8) &\Rightarrow y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_+ \rangle + b) \geq 1 \cdot y_i \\ &\Rightarrow y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_+ \rangle + b) \geq 1 \end{aligned} \tag{10}$$

$$\begin{aligned} (9) &\Rightarrow y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_- \rangle + b) \leq -1 \cdot y_i \\ &\Rightarrow y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_- \rangle + b) \geq 1 \end{aligned} \tag{11}$$

Ligning (11) ligner ligning (10), da observationen, der tilhører den negative klasse og skrives $\overrightarrow{x}_-$, medfører, at y_i er lig -1, som yderligere medfører, at ulighedstegnet

vendes, da der ganges med en negativ værdi på begge sider.

Både ligning (10) og (11) er ens og kan samles som

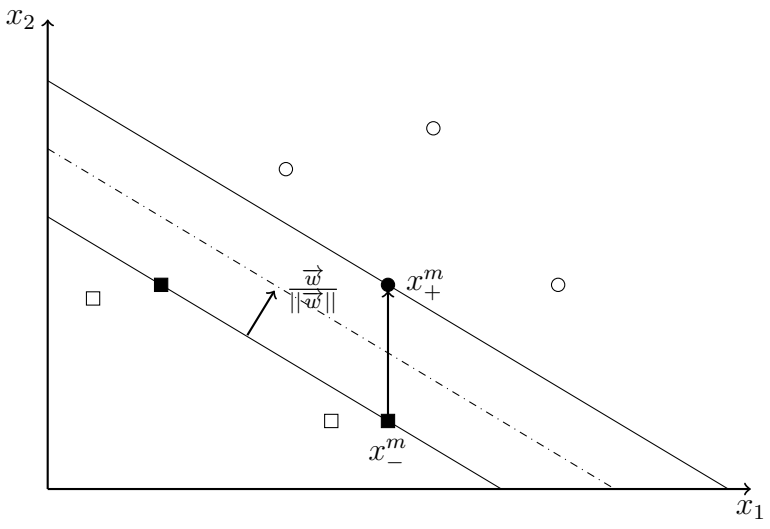
$$y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) \geq 1 \tag{12}$$

$$\Rightarrow \quad y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) - 1 \geq 0. \tag{13}$$

For enhver observation, der ligger på marginen, fås da

$$(13) \Rightarrow y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) - 1 = 0. \tag{14}$$

For at udregne bredden på den geometriske margin skal vi bruge to observationer, der ligger på hver sin side af marginen. Disse kalder vi nu henholdsvis $\overrightarrow{x}_+^m$ og $\overrightarrow{x}_-^m$. Hvis vi tager forskellen mellem disse to observationer, får vi en vektor, der går fra marginen på den ene side af hyperplanet til marginen på den anden side. En sådan vektor er illustreret i figur 6 herunder.m Vi er dog ikke sikre på, at det er en vektor, der er ortogonal på hyperplanet, dvs. vi ikke med sikkerhed ved, om det er den kortest mulige afstand mellem de to marginer.



Figur 6

Fordi vi ikke ved hvorvidt det er den kortest mulige afstand, er vi nødt til at prikke vektoren mellem x_+^m og x_-^m med en normal enhedsvektor, dvs. en vektor der er or-

togonal på hyperplanet, og som har længden 1, for dermed at udregne den kortest mulige længde mellem de to marginer. Vi har fra tidligere, at $\overrightarrow{w}$ er en normal vektor. Denne kan vi gøre til en normal enhedsvektor ved at dividere $\overrightarrow{w}$ med sin egen længde. Dette giver os, at bredden på marginen er givet ved

$$\left\langle (\overrightarrow{x}_+^m - \overrightarrow{x}_-^m) \cdot \frac{\overrightarrow{w}}{||\overrightarrow{w}||} \right\rangle \Rightarrow (\langle \overrightarrow{w} \cdot \overrightarrow{x}_+^m \rangle - \langle \overrightarrow{w} \cdot \overrightarrow{x}_-^m \rangle) \frac{1}{||\overrightarrow{w}||}. \tag{15}$$

Indsætter vi vores observation i den positive klasse liggende på marginen, dvs. $\overrightarrow{x}_+^m$, i ligning (14), der beskrev enhver observation på marginen, gælder følgende.

$$\begin{aligned} 1(\langle \overrightarrow{w} \cdot \overrightarrow{x}_+^m \rangle + b) - 1 &= 0 \\ \Rightarrow \langle \overrightarrow{w} \cdot \overrightarrow{x}_+^m \rangle &= 1 - b, \end{aligned} \tag{16}$$

og følgende på den anden side af marginen med $\overrightarrow{x}_-^m$

$$\begin{aligned} -1(\langle \overrightarrow{w} \cdot \overrightarrow{x}_-^m \rangle + b) - 1 &= 0 \\ \Rightarrow -\langle \overrightarrow{w} \cdot \overrightarrow{x}_-^m \rangle &= 1 + b \\ \Rightarrow \langle \overrightarrow{w} \cdot \overrightarrow{x}_-^m \rangle &= -1 - b. \end{aligned} \tag{17}$$

Vi indsætter ligning (16) og (17) i ligning (15) og får

$$(\langle \overrightarrow{w} \cdot \overrightarrow{x}_+^m \rangle - \langle \overrightarrow{w} \cdot \overrightarrow{x}_-^m \rangle) \frac{1}{||\overrightarrow{w}||} \Rightarrow ((1-b) - (-1-b)) \frac{1}{||\overrightarrow{w}||} = \frac{2}{||\overrightarrow{w}||}. \tag{18}$$

Af ligning (18) kan vi desuden udregne den geometriske margin M, som nødvendigvis må være halvdelen, dvs.

$$M = \frac{1}{2} \frac{2}{||\overrightarrow{w}||} = \frac{1}{||\overrightarrow{w}||}.$$

Dette er altså bredden af den geometriske margin, som vi ønsker at maksimere. At maksimere $\frac{1}{||\overrightarrow{w}||}$ svarer til at minimere $||\overrightarrow{w}||$ eller til at minimere $\frac{1}{2}||\overrightarrow{w}||^2$. Vi laver disse skridt for at nå frem til et kvadratisk optimeringsproblem, som vi også tidligere har sagt, vil vise sig at være nyttigt senere. Det kvadratiske optimeringsproblem bliver

$$\begin{aligned} &\text{min} && \frac{1}{2}||\overrightarrow{w}||^2 \\ &\text{ubb.} && y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) - 1 \geq 0 \qquad i = 1, ..., n. \end{aligned}$$

Teorien om marginerne som vi indtil nu har beskrevet, er baseret på teorien om den hårde margin, som vi nu kort vil beskrive. Efterfølgende vil vi introducere teorien om den bløde margin.

6.3.2.1 Hård margin

Med hård margin tillader vi ikke, at der kan ligge observationer mellem hyperplanet og marginen, som vi beskrev tidligere i afsnittet, hvor vi kom frem til ligning (8) og ligning (9). Observationer, der ligger på marginen, er altså de observationer, der ligger nærmest hyperplanet. På den måde kan vi sige, at det er den hårde margin, der bruges til klassifikationsproblemer med lineært separable observationer.

Observationerne omkring hyperplanet med hård margin skal derfor opfylde

$$\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b \begin{cases} \geq 1 & for\ y_i = 1 \\ \leq -1 & for\ y_i = -1 \end{cases} .$$

Fordi data er lineært separabelt med den hårde margin, vil der ikke være tilfælde, der opfylder

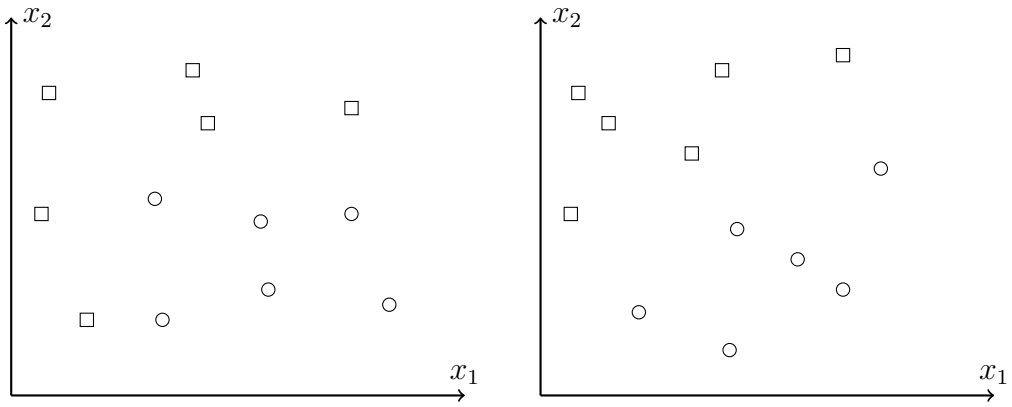
$$-1 < \langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b < 1, \qquad i = 1, ..., n.$$

Med ikke-lineært separabelt data vil der ikke være nogen mulig løsning for den hårde margin. Her kan vi i stedet bruge teori om den bløde margin, som vi nu vil introducere.

6.3.2.2 Blød margin

Det er ofte ikke muligt at separere observationer lineært, og vi vil derfor her beskrive den metode, som SVM anvender til at løse denne udfordring. Lad os først se på figur 7 herunder. Figuren illustrerer to rum med observationer fordelt i to klasser, igen hvor den ene klasse er symboliseret ved firkanter, og den anden klasse er ved cirkler. Rummet til venstre viser et eksempel på ikke-lineært separable observationer, og

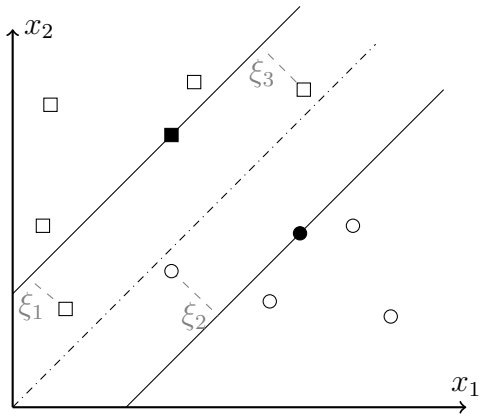
rummet til højre er et eksempel på lineært separable observationer. Dette vil sige, at observationerne i rummet til højre i figuren kan separeres ved et lineært hyperplan. Det kan observationerne i rummet til venstre til gengæld ikke.



Figur 7

Når observationerne ikke er lineært adskillelige, og data derfor ikke er lineært separabelt, kan vi ikke bruge teorien fra hård margin til at finde det optimale separerende hyperplan, medmindre vi konstruerer en margin, der er blød.

Med blød margin kan vi tillade, at der kan ligge observationer mellem hyperplanet og marginen. Et lineært separerende hyperplan med blød margin er illustreret i figur 8 herunder.



Figur 8

Enhver observation, der ligger mellem hyperplanet og marginen, opfylder

$$\langle \overrightarrow{w} \cdot \overrightarrow{x} \rangle + b = c, \tag{19}$$

hvor $-1 < c < 1$. Sådan en observation tillægger vi en slackværdi.

Vi introducerer nu *hinge loss-funktionen*, som vurderer slackværdien for alle observationer i rummet. Hinge loss-funktionen er defineret ved

$$\xi_i = \max\big(0, 1 - y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b)\big), \quad i = 1, ..., n. \tag{20}$$

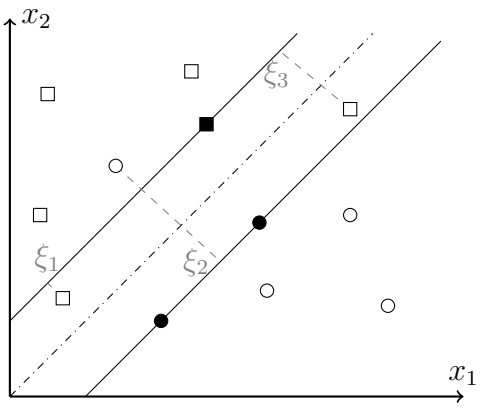
Hvis en observation er klassificeret korrekt, så vil y_i og $(\langle \overrightarrow{w}_i \cdot \overrightarrow{x}_i \rangle + b)$ have samme fortegn. Dette medfører, at $1 - y_i(\langle \overrightarrow{w}_i \cdot \overrightarrow{x}_i \rangle + b)$ ikke kan blive større end 1. Ligeledes medfører det, at ξ_i er ikke-negativ, dvs. $\xi_i \geq 0$, da hinge-loss funktionen altid vælger positivdelen. For observationer, der er klassificeret korrekt og dermed ligger på den korrekte side af marginen, er det klart, at den respektive slackværdi vil være $\xi_i = 0$.

Når vi tillader observationer at ligge mellem marginen og det separerende hyperplan, og observationen stadig er klassificeret korrekt, dvs. at den ligger på den korrekte side af hyperplanet, vil hinge loss-funktionen opfylde $0 < \xi_i < 1$.

Med en blød margin vil det også være muligt, at observationen ligger på modsatte side af hyperplanet og sågar ligger længere ude end den modsatte side af marginen. I sådanne tilfælde siger vi, at observationen er *misklassificeret*.

Misklassificering

Misklassificering kan opstå, når vi har blød margin, da den bløde margin vil tillade observationer at ligge på den anden side af hyperplanet.



Figur 9

I figur 9 har vi en blød margin, og i dette tilfælde ligger to af observationerne på den modsatte side af hyperplanet i forhold til deres klasse. Disse to observationer kalder vi misklassificerede observationer. Observationen med ξ_1 er dog stadig klassificeret korrekt.

For en tilfældig misklassificeret observation gælder der

$$y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) < 0, \qquad i = 1, ..., n,$$

da y_i og $(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b)$ vil have forskellige fortegn.

Ser vi på Hingeloss-funktionen for en misklassificeret observation, så ser vi altså på

$$\xi_i = \max\big(0, 1 - y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b)\big), \qquad i = 1, ..., n,$$

hvor $y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) < 0$. Det medfører at slackværdien $\xi_i > 1$ for enhver misklassificeret observation.

Uanset om observationen er korrekt klassificeret eller misklassificeret, kan vi nu tilføje slackværdien til ligning (12). Vi får således bibetingelsen

$$y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) \geq 1 - \xi_i, \qquad i = 1, ..., n.$$

Den totale slack kan findes ved $\sum_{i=1}^{n} \xi_i$, som vi senere ønsker at bruge, når vi skal finde det optimale hyperplan. Af den grund må observationer med slack også have

en indflydelse på marginen. Enhver observation, der indeholder slack, er dermed en supportvektor. Det medfører, at observationer med slack har en α_i værdi større end 0.

Vi ønsker stadig at løse problemet for marginen, hvor vi minimerer $\frac{1}{2}||\overrightarrow{w}||^2$. Til problemet tilføjer vi den totale slack, som vi ligeledes ønsker at minimere. Dog tillægger vi først slacken en vægt C, der beskriver, hvor hårdt vi straffer observationer, der har slack. Jo større C, des mere straffer vi misklassificerede observationer. Det betyder også, at når C stiger, så vil $\sum_{i=1}^{n} \xi_i$ falde, da man intuitivt vil være mindre tilbøjelig til at tillade observationer at have slack.

Det giver derfor også mening, at $C = \infty$ vil medføre hård margin, hvor vi ikke kan tillade slack, da enhver slack vil straffes uendeligt meget. Vi tillader desuden ikke, at $C = 0$, da slacken derved ikke vil have betydning for marginen, ligesom vi heller ikke tillader at $C < 0$, da dette betyder, at vi ville belønne slack.

Ligeledes betyder det illustrativt, at hvis man lader C gå mod nul, vil man kunne opnå en bredere margin dog med en forventning om flere supportvektorer. Man er derfor nødt til at justere C. Det kan vi gøre ved at *tune* parameteret C. Tuning er en proces, vi beskriver i afsnit 8.3.

Vi kan nu opstille optimeringsproblemet for blød margin.

$$\begin{aligned} \min \quad & \tfrac{1}{2}||\overrightarrow{w}||^2 + C\sum_{i=1}^{n} \xi_i \\ \text{ubb.} \quad & y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \end{aligned}$$

for $i = 1, ..., n$.

Selvom data ikke kan adskilles lineært, er det muligt at foretage transformationer af data, således at data lettere kan adskilles af et lineært hyperplan i det rum, hvor data er transformeret. I SVM bruges *kerner* til at foretage disse transformationer.

6.4 Kerneteori

Vi ved fra statistik, at det ofte kan være nyttigt at ændre repræsentationen af data. Det er eksempelvis ofte nyttigt at tage logaritmen til de beskrivende variable. Kerner bruges netop til at ændre den måde, hvorpå data repræsenteres ved at projicere data ind i et nyt rum - oftest et rum med en højere dimension, således at data kan klassificeres lineært i det nye rum. I dette afsnit vil vi beskrive kerner, hvorfor de er brugbare, hvad definitionen på kerner er, og hvilke betingelser vi har i forbindelse med kerner.

Vi ændrer repræsentationen af data $\overrightarrow{x} \in X$, så

$$\overrightarrow{x} = (x_1, ..., x_n) \mapsto \overrightarrow{\phi}(\overrightarrow{x}) = (\phi_1(\overrightarrow{x}), \phi_2(\overrightarrow{x}), ..., \phi_N(\overrightarrow{x})).$$

Projiceringen kan skrives som

$$F = \{\overrightarrow{\phi}(\overrightarrow{x}) | \overrightarrow{x} \in X\}.$$

En kerne er en funktion K, der foretager projiceringen implicit ved en funktion af det indre produkt af observationer i X. Funktionen giver altså det samme resultat, som hvis transformationen blev udregnet eksplicit. Kernen kan vi dermed skrive som

$$\forall\ \overrightarrow{x_i}, \overrightarrow{x_j} \in X : K(\overrightarrow{x_i}, \overrightarrow{x_j}) = \langle \overrightarrow{\phi}(\overrightarrow{x_i}) \cdot \overrightarrow{\phi}(\overrightarrow{x_j}) \rangle, \quad i, j = 1, .., n. \tag{21}$$

Kompleksiteten af beslutningsfunktionen $f(\overrightarrow{x})$, som skal trænes, afhænger af, hvordan træningsdata S er repræsenteret. Ved at repræsentere træningsdata på en ny måde er det muligt at simplificere træningen af beslutningsfunktionen. Ofte ønskes det, at denne træning skal være lineær, hvilket netop er tilfældet ved SVM.

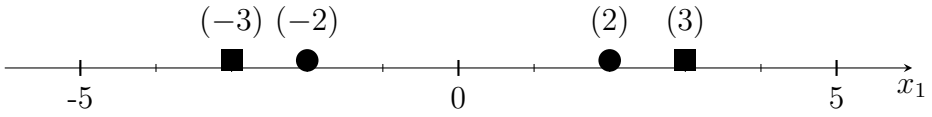
Vi kan bruge vores projicering til at beholde lineariteten af beslutningsfunktionen fra kapitel 6.3 ligning (6) og samtidig anvende ikke-lineære kombinationer af vores forklarende variable ved at skrive beslutningsfunktionen som

$$f(\overrightarrow{x}) = \sum_{i=1}^{n} w_i \overrightarrow{\phi}(x_i) + b.$$

Eksempel 1.

Eksemplet her illustrerer, hvordan vi kan bruge en projicering $\overrightarrow{\phi}$ til at adskille observationer, der ikke er lineært separable i deres originale rum, med et lineært hyperplan i det nye rum.

Betragt et træningssæt med en klasse bestående af endimensionale observationer $(\overrightarrow{x}_1, \overrightarrow{x}_2) = \{(2), (-2)\}$ samt en anden klasse med observationerne $(\overrightarrow{x}_3, \overrightarrow{x}_4) = \{(3), (-3)\}$. Vi skriver træningssættet som $S = (\overrightarrow{x}_1, \overrightarrow{x}_2, \overrightarrow{x}_3, \overrightarrow{x}_4)$. Vi illustrerer først observationerne i det endimensionale rum i figuren herunder.

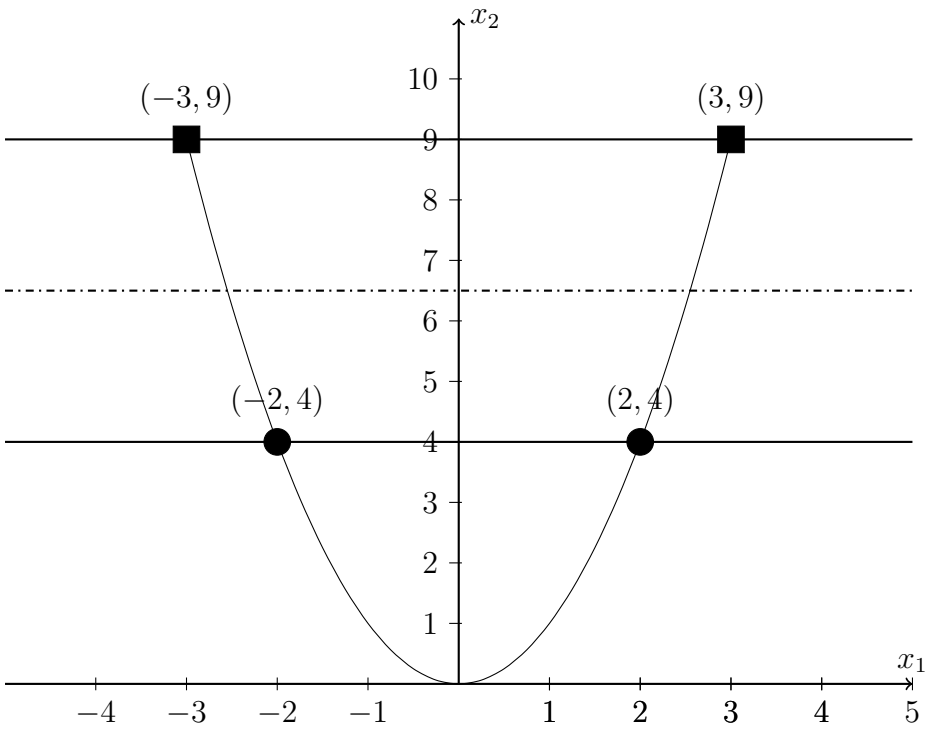


Figur 10

I figuren kan observationerne ikke adskilles ved et lineært separerende hyperplan. Dette er fordi, hyperplanet i det endimensionale rum er et punkt, hvilket vores observationer ikke kan adskilles med.

Fra figur 10 til 11 ændres repræsentationen af data, således at det bliver muligt at foretage den simple lineære klassifikation ved et hyperplan illustreret ved den stiplede linje. Her beskrives transformationen ved

$$\overrightarrow{\phi}(\overrightarrow{x}_i) = (\phi_1(\overrightarrow{x}_i), \phi_2(\overrightarrow{x}_i)) = (\overrightarrow{x}_i, \overrightarrow{x}_i^2), \quad i = 1, 2, 3, 4.$$



Figur 11

□

Hvis repræsentationen af data kan klassificeres lineært i et rum med n dimensioner, vil data også altid kunne klassificeres lineært i et rum med m dimensioner, hvor $m > n$. Det er altså aldrig en ulempe i forhold til adskillelse af observationer at øge dimensionen af det rum, der projiceres ind i. Vi risikerer dog at øge beregningskraften samt at få problemer med generaliseringen, som vi vil beskrive i afsnit 6.5. Problemet med den ekstra beregningskraft kan vi løse ved *kernetricket*.

6.4.1 Kernetricket

En lineær learning machine træner beslutningsfunktionen ved en lineær kombination af de forklarende variable. Problemet med lineære learning machines, som vi så ovenfor, er, at det kan være umuligt at klassificere lineært i det originale rum.

Kernetricket består i at udskifte indre produkter med en kernefunktion defineret som ovenfor. Vi vil senere vise at data udelukkende vil indgå i beslutningsfunktionen som indre produkter. Kernefunktionen sørger for, at data projiceres ind i et rum af en højere dimension. Vi skal se, at dette bliver gjort implicit, hvormed vi undgår at anvende beregningskraft proportionalt med antallet af dimensioner i det nye rum, der projiceres ind i. Udregningerne vil i stedet foregå i det originale inputrum. Dette vil vi illustrere i eksempel 4.

Da kernetricket transformerer observationerne, bliver det muligt at træne beslutningsfunktionen lineært men med ikke-lineære kombinationer af de forklarende variable i én samlet beregning. Dette er nyttigt, fordi sandsynligheden for en lineær separation af data er proportional med antallet af dimensioner, for det rum observationerne projiceres ind i. Der vil i teorien altid findes et rum, hvori data kan adskilles lineært.

Ved den populære kerne *radial basis function kernel*, herefter RBF-kerne, anvender man netop transformationer til et uendelig-dimensionalt rum. I afsnit 6.4.3.2 vil vi illustrere, at der stadig eksisterer en endelig værdi for kernefunktionen, når vi opererer med uendelig-dimensionale rum, og i eksempel 5 vil vi vise, at det kan lade sig gøre i forhold til den beregningskraft, problemet vil kræve.

Kunsten er nu at finde kerner, altså funktioner, der transformerer implicit. SVM gør brug af forskellige kerner, der hver især løser problemet med uadskillelige observationer bedst ved forskellige situationer. Fælles for disse funktioner er, at de opfylder egenskaberne for kerner. Disse egenskaber er givet ved Mercers sætning.

6.4.2 Mercers sætning

Mercers sætning siger, at hvis vi lader X være en kompakt delmængde af $\mathbb{R}^P$, så er funktionen $K = X \times X \to \mathbb{R}$ en kerne, når følgende er opfyldt.

1. K er symmetrisk dvs. $K(\overrightarrow{x}_i, \overrightarrow{x}_j) = K(\overrightarrow{x}_j, \overrightarrow{x}_i)$, for $i, j = 1, ...n$.

2. Matricen
$$\mathbf{K} = \begin{bmatrix} K(\overrightarrow{x}_1, \overrightarrow{x}_1) & \cdots & K(\overrightarrow{x}_1, \overrightarrow{x}_n) \\ \vdots & \ddots & \vdots \\ K(\overrightarrow{x}_n, \overrightarrow{x}_1) & \cdots & K(\overrightarrow{x}_n, \overrightarrow{x}_n) \end{bmatrix},$$
som vi kalder kernens Gram-matrix, er en $n \times n$ positiv semi-definit matrice $\forall 1, ..., n \in \mathbb{N}$, dvs.
$$\overrightarrow{V}^T \mathbf{K} \overrightarrow{V} = \sum_{i,j=1}^{n} v_i K(\overrightarrow{x}_i, \overrightarrow{x}_j) v_j \geq 0 \qquad \forall v_i \in \mathbb{R}.$$

Hvis de to ovenstående kriterier gælder, så findes der et $\overrightarrow{\phi} : X \to Y$, hvor Y er en kompakt delmængde af $\mathbb{R}^O$, så
$$\langle \overrightarrow{\phi}(\overrightarrow{x}) \cdot \overrightarrow{\phi}(\overrightarrow{y}) \rangle = K(\overrightarrow{x}, \overrightarrow{y}),$$
hvis og kun hvis
$$\int_{X \times X} f(\overrightarrow{x}) K(\overrightarrow{x}, \overrightarrow{y}) f(\overrightarrow{y}) d\overrightarrow{x} d\overrightarrow{y} \geq 0, \tag{22}$$
$\forall f : X \to \mathbb{R}$, hvor $\int f(\overrightarrow{x})^2 d\overrightarrow{x} < \infty$.

Dette kan vi vise ved et eksempel. Hvis vi har en funktion,
$$f_\delta(\overrightarrow{x}) = \sum_{i=1}^{n} f_{i,\delta}(\overrightarrow{x}),$$
hvor
$$f_{i,\delta}(\overrightarrow{x}) = v_i \cdot \frac{1}{\delta} \cdot \mathbf{1}_{]x_i - \frac{\delta}{2}, x_i + \frac{\delta}{2}[},$$

hvor **1** er en indikator funktion, så vil kravet blive

$$\begin{aligned} 0 &\leq \int f(\overrightarrow{x})k(\overrightarrow{x},\overrightarrow{y})f(\overrightarrow{y})d\overrightarrow{x}\,d\overrightarrow{y} \\ &= \sum_{i,j=1}^{n} \int f_{i,\delta}(\overrightarrow{x})K(\overrightarrow{x},\overrightarrow{y})f_{j,\delta}(\overrightarrow{y})d\overrightarrow{x}\,d\overrightarrow{y} \\ &= \sum_{i,j=1}^{n} v_i v_j \cdot \frac{1}{\delta^2} \cdot \int_{\overrightarrow{x}\in]x_i-\frac{\delta}{2},\, x_i+\frac{\delta}{2}[} \int_{\overrightarrow{y}\in]y_j-\frac{\delta}{2},\, y_j+\frac{\delta}{2}[} K(\overrightarrow{x},\overrightarrow{y})d\overrightarrow{x}\,d\overrightarrow{y}. \end{aligned} \tag{23}$$

Hvis vi nu lader $\delta \to 0$, fås

$$(23) \Rightarrow \overrightarrow{V}^T \mathbf{K} \overrightarrow{V}.$$

Er $\overrightarrow{V}^T \mathbf{K} \overrightarrow{V} \leq 0$, så må integralet $\int f(\overrightarrow{x})K(\overrightarrow{x},\overrightarrow{x})f(\overrightarrow{y})d\overrightarrow{x}\,d\overrightarrow{y} \leq 0$, hvilket betyder, at $K(\overrightarrow{x},\overrightarrow{y}) \neq \langle \overrightarrow{\phi}(\overrightarrow{x}) \cdot \overrightarrow{\phi}(\overrightarrow{y}) \rangle$. Derfor kan vi konkludere, at **K** skal være positiv definit.

Kerner har som nævnt mulighed for at transformere til et uendelig-dimensionalt rum. Da disse transformationer sker ved indre produkter, er det vigtigt, at vi opererer i et rum, der tillader dette. Dette tillader *Hilbertrummet*.

6.4.3 Hilbertrum

Vi vil i dette afsnit redegøre for, hvad et Hilbertrum er, nævne dets egenskaber og forklare de vigtigste årsager til, at vi ønsker at operere i Hilbertrum.

Hilbertrum generaliserer euklidiske rum, idet et Hilbertrum kan operere med uendelige dimensionale rum. Denne egenskab er nyttig, idet vi ved, at vi med kernetricket kan ændre repræsentationen af data, og at jo højere dimensionen af den nye repræsentation er, jo nemmere er data at separere.

Den anden grund, til at vi er interesseret i at arbejde i et Hilbertrum, er, at vi senere viser, at data kun vil indgå i objektfunktionen for optimeringsproblemet som indre produkter. Et Hilbertrum tilføjer nemlig det indre produkt som en operation

til det euklidiske rum, altså produktet mellem to vektorer. Denne operation skal nu overholde nogle egenskaber, som vi vil nævne nedenfor.

6.4.3.1 Egenskaber og operationer

Da Hilbertrummet generaliserer det euklidiske rum, er dette også et vektorrum. Et vektorrum er en ikke-tom mængde V af vektorer, hvori der er defineret to operationer kaldet *addition* og *multiplikation* af *skalarer*, som er underlagt ti aksiomer. De ti aksiomer må gælde for alle vektorer $\overrightarrow{u}$, $\overrightarrow{v}$, og $\overrightarrow{w} \in V$ og for alle skalarer $c, d \in \mathbb{R}$.

1. $\overrightarrow{u} + \overrightarrow{v} \in V$
2. $\overrightarrow{u} + \overrightarrow{v} = \overrightarrow{v} + \overrightarrow{u}$
3. $(\overrightarrow{u} + \overrightarrow{v}) + \overrightarrow{w} = \overrightarrow{u} + (\overrightarrow{v} + \overrightarrow{w})$
4. $\overrightarrow{0}$ er en nul-vektor i V således at $\overrightarrow{u} + \overrightarrow{0} = \overrightarrow{u}$
5. $\forall\ \overrightarrow{u} \in V$ findes der en vektor $(-\overrightarrow{u}) \in V$, sådan at $\overrightarrow{u} + (-\overrightarrow{u}) = \overrightarrow{0}$
6. $\overrightarrow{u} \cdot c \in V$
7. $c(\overrightarrow{u} + \overrightarrow{v}) = c \cdot \overrightarrow{u} + c \cdot \overrightarrow{v}$
8. $(c + d) \cdot \overrightarrow{u} = c \cdot \overrightarrow{u} + d \cdot \overrightarrow{u}$
9. $c(d \cdot \overrightarrow{u}) = (c \cdot d) \cdot \overrightarrow{u}$
10. $1 \cdot \overrightarrow{u} = \overrightarrow{u}$

Ligesom vektorrummet har to operationer, har Hilbertrum yderligere det indre produkt som operation. Denne operation skal nu opfylde betingelserne konjugeret symmetri, linearitet/antilinearitet, positiv definithed og afstand ved indre produkter, som beskrives herunder.

1) Konjugeret symmetri
Der skal gælde at

$$\langle x_1, x_2 \rangle = \langle x_2, x_1 \rangle^{\sim},$$

hvor $^{\sim}$ er det, som vi kalder det komplekse konjugat. Det komplekse konjugat er

det, der tager den imaginære del af en vektor, og vender det om. Et eksempel kunne være

$$v = a + bi,$$

hvor det komplekse konjugat vil være

$$v^{\sim} = a - bi.$$

2a) Linearitet mht. 1. vektor i det indre produkt

$$\langle ax_1 + bx_2, x_3\rangle = a\langle x_1, x_3\rangle + b\langle x_2, x_3\rangle.$$

Så hvis vi tager det indre produkt mellem to vektorer, hvor første element er skaleret med to skalarer, vil de komme ud urørt.

2b) Antilinearitet mht. 2. vektor i det indre produkt

Hvis vi derimod tager skalarene på andet element af det indre produkt, vil vi være nødt til at tage de komplekse konjugater af skalarene. Så

$$\langle x_1, ax_2 + bx_3\rangle = a^{\sim}\langle x_1, x_2\rangle + b^{\sim}\langle x_1, x_3\rangle.$$

3) Positiv semi-definit

Det indre produkt af en vektor med den selv skal være ikke-negativ, dvs.

$$\langle v, v\rangle = ||v||^2 \geq 0,$$

hvor der kun er lighed, når vektoren er nulvektoren. Dette er også kendt som positiv semi-definithed.

Et vektorrum, der opfylder denne betingelse, er kaldet et normeret vektorrum.

4) Afstand

Afstanden mellem to punkter i et Hilbertrum udregnes ved det indre produkt,

$$|\overrightarrow{x}_2 - \overrightarrow{x}_1| = \sqrt{\langle(\overrightarrow{x}_2 - \overrightarrow{x}_1)\cdot(\overrightarrow{x}_2 - \overrightarrow{x}_1)\rangle} \in \mathbb{R}^+.$$

Et vektorrum, der opfylder denne betingelse, kaldes et metrisk rum.

Yderligere betingelser

Ud over at Hilbertrummet skal opfylde de samme aksiomer som et lineært vektorrum og indeholde et indre produkt, skal det også opfylde tre yderligere betingelser kaldet separabilitet, fuldstændighed og uendelig dimensionalitet.

a) Separabilitet

Et rum er separabelt, hvis det indeholder en delmængde, der er tællelig og kompakt. De reelle tal er et eksempel på dette.

Eksempel 2.

De reelle tal $\mathbb{R}$ har de rationelle tal $\mathbb{Q}$ som delmængde. De rationelle tal er tællelige, fordi de som bekendt er defineret som forholdet mellem heltal og naturlige tal altså a/b, hvor a og b begge er tællelige.

De rationelle tal er også kompakte, hvilket betyder, at alle elementer i det topologiske rum tilhører mængden eller kan approksimeres uendelig præcist. Eulers tal e er et eksempel på et tal i det topologiske rum, som kan approksimeres med rationelle tal med uendelig høj præcision.

$$e \approx 1 + \sum_{i=1}^{n} \tfrac{1}{i!}, \qquad n \to \infty.$$

Eksempelvis for $n = 1$

$$1 + \sum_{i=1}^{1} \frac{1}{1} = 2$$

og $n = 5$

$$1 + \sum_{i=1}^{5} \frac{1}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5} = 2.717.$$

□

b) Fuldstændighed

Et rum siges at være fuldstændigt, hvis enhver Cauchyfølge i rummet konvergerer i

rummet. En Cauchyfølge er en følge, hvor elementerne konvergerer mod hinanden, efterhånden som følgen går mod uendelig. Dette kan skrives ved følgende

$$\lim_{m,n\to\infty} |x_m - x_n| = 0.$$

Et normeret vektorrum, der er fuldstændigt, kaldes et Banach rum, og alle Hilbertrum er Banach rum.

c) Uendelig dimensionalt

For at et rum skal være af den n'te dimension, skal det indeholde n lineært uafhængige vektorer. Et Hilbertrum har n lineære uafhængige vektorer for alle $n \in \mathbb{N}$.

Et Hilbertrum kan altså betegnes som et fuldstændigt normeret metrisk lineært indre produkts vektorrum, som kan arbejde i den uendelige dimension.

6.4.3.2 Dimensionalitet

SVM anvender som nævnt kernetricket, der ændrer repræsentationen af træningsdata ved at transformere dette implicit ind i et potentielt uendelig-dimensionalt rum. Et Hilbertrum kan være enten endelig-dimensionalt eller uendelig-dimensionalt.

For at et indre produkt eksisterer i et Hilbertrum af uendelig dimension, skal følgende integrale konvergere for funktioner på komplekse værdier, dvs.

$$\langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle = \int_{-\infty}^{\infty} \overrightarrow{x_i} \sim \overrightarrow{x_j} dx < \infty, \qquad i, j = 1, .., n.$$

Dette er opfyldt, når integralet af det kvadrerede $\langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle$ er endeligt, dvs. at følgende konvergerer

$$\int_{-\infty}^{\infty} ||\overrightarrow{x}||^2 dx < \infty.$$

Da kernetricket gør brug af det indre produkt, er det essentielt, at denne betingelse er opfyldt, da det afgør eksistensen af det indre produkt.

Muligheden for at arbejde i et uendelig-dimensionalt rum vil vise sig nyttigt, når vi nu vil beskrive nogle af de populære kerner, hvor en af dem er RBF-kernen, der arbejder i et uendelig-dimensionalt rum.

6.4.4 Populære kerner

I dette afsnit vil vi se nærmere på tre af de mest populære kerner nemlig den polynomiske, lineære og RBF-kernen, hvilket også er de kerner, vi vil bruge til at udvikle vores model til prædiktion af konkurser i analysen.

6.4.4.1 Den polynomiske kerne

Den første kerne vi skal se på, er den polynomiske kerne og er defineret ved

$$K(\overrightarrow{x_i}, \overrightarrow{x_j}) = (\gamma\langle\overrightarrow{x_i} \cdot \overrightarrow{x_j}\rangle + c)^d, \qquad d \in \mathbb{N},\ \gamma, c \in \mathbb{R}^+, i, j = 1, ...n, \tag{24}$$

hvor $c = 0$ for homogene, og $c > 0$ for inhomogene polynomiske kerner. I den teoretiske litteratur er γ sat lig 1.

Når vi implementerer den polynomiske kerne i programmet R, er c, d og γ de parametre, der skal vælges.

En inhomogen polynomisk kerne af anden grad, dvs. $d = 2$, er eksempelvis givet ved

$$\begin{aligned} K(\overrightarrow{x_i}, \overrightarrow{x_j}) &= (\gamma\langle\overrightarrow{x_i} \cdot \overrightarrow{x_j}\rangle + c)^2 \\ &= \gamma^2\langle\overrightarrow{x_i} \cdot \overrightarrow{x_j}\rangle^2 + c^2 + 2\gamma c\langle\overrightarrow{x_i} \cdot \overrightarrow{x_j}\rangle. \end{aligned} \tag{25}$$

Eksempel 3.

Betragt en polynomisk kerne med $\gamma = 1$, $c = 0$ og $d = 2$ og to observationer $\overrightarrow{x}, \overrightarrow{y} \in \mathbb{R}^2$, hvor $\overrightarrow{x} = (x_1, x_2)$ *og* $\overrightarrow{y} = (y_1, y_2)$. Så er,

$$K(\overrightarrow{x}, \overrightarrow{y}) = \langle \overrightarrow{x} \cdot \overrightarrow{y} \rangle^2 = x_1^2 y_1^2 + 2x_1 y_1 x_2 y_2 + x_2^2 y_2^2.$$

Vi vil nu finde en transformation $\overrightarrow{\phi}$, så vi kan skrive

$$K(\overrightarrow{x}, \overrightarrow{y}) = \langle \overrightarrow{\phi}(\overrightarrow{x}) \cdot \overrightarrow{\phi}(\overrightarrow{y}) \rangle.$$

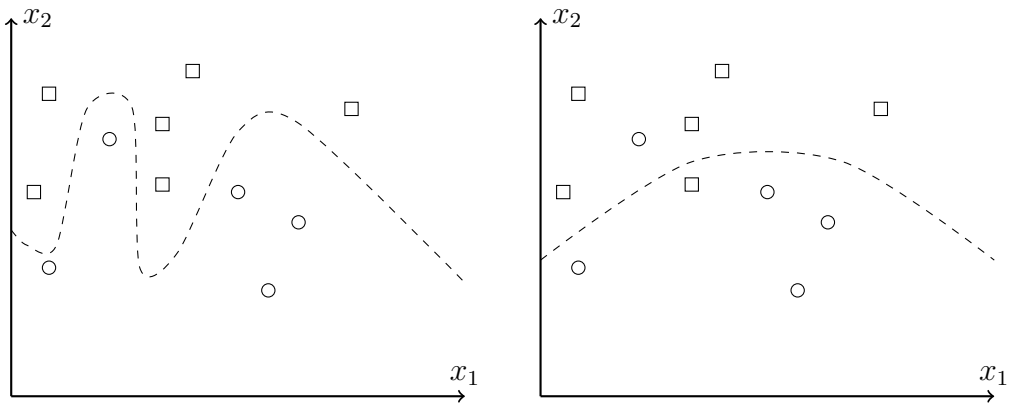
Et $\overrightarrow{\phi}$, der opfylder dette, er

$$\begin{aligned} \overrightarrow{\phi}(\overrightarrow{x}) &= (x_1^2, \sqrt{2}x_1 x_2, x_2^2) \\ \Rightarrow \langle \overrightarrow{\phi}(\overrightarrow{x}) \cdot \overrightarrow{\phi}(\overrightarrow{y}) \rangle &= x_1^2 y_1^2 + 2x_1 y_1 x_2 y_2 + x_2^2 y_2^2. \end{aligned}$$

Vi har her en transformation, der tager observationer fra $\mathbb{R}^2$ og projicerer dem ind i $\mathbb{R}^3$. Dette er nyttigt, da vi ved at sandsynligheden, for at data bliver lineært adskilligt, stiger med dimension af det rum, der projiceres ind i.

□

Den polynomiske kerne adskiller data ved et hyperplan, der, når det projiceres tilbage i det originale inputrum, er et d-dimensionalt polynomium som illustreret nedenfor.



Figur 12

På ovenstående figur adskiller femtegradspolynomiet (venstre) observationerne perfekt, mens andensgradspolynomiet (højre) ser ud til at være en model, der kan generaliseres bedre til ny data. En for høj værdi af d kan altså gøre, at vi får en model, der er for specifik med hensyn til data. Dette fænomen kaldes *overfitting* og vil blive beskrevet nærmere i afsnit 6.5.

I afsnit 8.1.3 vil vi se, at den polynomiske kerne har en udfordring, når den skal implementeres numerisk, idet vi kan komme ud for, at algoritmen ikke konvergerer.

Vi vil nu bevise, at den polynomiske kerne er en kerne ifølge Mercers sætning.

Bevis

Vi vil først udføre beviset for den polynomiske kerne med $d = 2$ defineret ved ligning (25) ovenfor. Jf. Mercers sætning er en funktion en kerne, hvis funktionen er symmetrisk, samt at dens tilhørende Gram-matrix er positiv semi-definit.

En matrix $\mathbf{K}$ er positiv semidefinit hvis og kun hvis

$$\forall \overrightarrow{\zeta} \in \mathbb{R}^n : \overrightarrow{\zeta}^T \mathbf{K} \overrightarrow{\zeta} \geq 0.$$

Ser vi på den polynomiske kerne af anden grad fra ligning (25), kan vi opdele denne i funktionerne $J_1(\overrightarrow{x_i}, \overrightarrow{x_j}) = \gamma^2 \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle$, $J_2(\overrightarrow{x_i}, \overrightarrow{x_j}) = c^2$ og $J_3(\overrightarrow{x_i}, \overrightarrow{x_j}) = 2\gamma c \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle$. Per definition er funktionen $\langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle$ en kerne, idet det er et indre produkt, som nødvendigvis må være en kerne. For at bevise at J_1 er en kerne, skal vi altså bevise, at en konstant ganget på en kerne er en kerne, samt at produktet af to kerner også er en kerne. J_2 er en kerne, hvis vi kan vise, at en konstant er en kerne. Vi kan vise, at J_3 er en kerne, hvis en positiv konstant ganget på en kerne også er en kerne. Til sidst mangler vi blot at bevise, at summen af to kerner er en kerne.

Vi kan fuldføre beviset, ved at bevise at følgende fire funktioner er kerner, hvor K_1 og K_2 er kerner, $f : X \to \mathbb{R}, X \subseteq \mathbb{R}^n$, $\overrightarrow{\zeta} \in \mathbb{R}^n$, og konstanten a er defineret på samme måde som c og γ ovenfor, nemlig ved $a \in \mathbb{R}^+$.

1) $K(\overrightarrow{x_i}, \overrightarrow{x_j}) = K_1(\overrightarrow{x_i}, \overrightarrow{x_j}) + K_2(\overrightarrow{x_i}, \overrightarrow{x_j})$

For at tjekke om funktionens Gram-matrix er positiv semi-definit, prikker vi med $\overrightarrow{\zeta}$ på hver side

$$\overrightarrow{\zeta}^T(\mathbf{K_1} + \mathbf{K_2})\overrightarrow{\zeta} = \overrightarrow{\zeta}^T\mathbf{K_1}\overrightarrow{\zeta} + \overrightarrow{\zeta}^T\mathbf{K_2}\overrightarrow{\zeta} \geq 0,$$

hvor sidste ulighed kommer af, at $\mathbf{K_1}$ og $\mathbf{K_2}$ er defineret til at være Gram-matricer for kerner, og dermed er positiv semi-definitte. Da to symmetriske matricer lagt sammen nødvendigvis må være symmetrisk, er Gram-matricens funktion det også, hvormed denne er en kerne.

2) $K(\overrightarrow{x_i}, \overrightarrow{x_j}) = aK_1(\overrightarrow{x_i}, \overrightarrow{x_j})$

Ved samme fremgangsmåde kan vi skrive

$$\overrightarrow{\zeta}^T a\mathbf{K_1}\overrightarrow{\zeta} = a\,\overrightarrow{\zeta}^T\mathbf{K_1}\overrightarrow{\zeta}^T \geq 0,$$

da $a \in \mathbb{R}^+$. En faktor ganget på en matrix skalerer blot matricens elementer, hvormed symmetrien bibeholdes. Funktionen er dermed en kerne.

3) $K(\overrightarrow{x_i}, \overrightarrow{x_j}) = K_1(\overrightarrow{x_i}, \overrightarrow{x_j})K_2(\overrightarrow{x_i}, \overrightarrow{x_j})$

Vi definerer først

$$k_{ij}^p = K_p(\overrightarrow{x}_i, \overrightarrow{x}_j), \qquad p = 1, 2, \;\; i, j = 1, ..., n$$

og dermed

$$\mathbf{K_p} = \begin{bmatrix} k_{11}^p & \cdots & k_{1n}^p \\ \vdots & \ddots & \vdots \\ k_{n1}^p & \cdots & k_{nn}^p \end{bmatrix},$$

og definerer yderligere

$$\mathbf{E} = \begin{bmatrix} k_{11}^1 k_{11}^2 & \cdots & k_{1n}^1 k_{1n}^2 \\ \vdots & \ddots & \vdots \\ k_{n1}^1 k_{n1}^2 & \cdots & k_{nn}^1 k_{nn}^2 \end{bmatrix}.$$

Vi søger altså at bevise, at

$$\overrightarrow{\zeta}^T \mathbf{E} \overrightarrow{\zeta} = \sum_{i,j=1}^{n} \zeta_i \zeta_j k_{ij}^1 k_{ij}^2 \geq 0,$$

samt at $\mathbf{E}$ er symmetrisk.

k_{ij}^p er symmetrisk, da hvert af disse elementer kommer fra en symmetrisk funktion nemlig K_p. Dermed er $k_{ij}^1 k_{ij}^2 = k_{ji}^1 k_{ji}^2$, hvormed $\mathbf{E}$ er symmetrisk.

Vi udnytter at $\mathbf{K_1}$ og $\mathbf{K_2}$ er positiv semi-definitte og symmetriske, idet de er kerner. Fra lineær algebra ved vi, at enhver symmetrisk positiv semi-definit matrix kan faktoriseres ved Cholesky dekompositionen

$$\mathbf{K_p} = \mathbf{R_p}^T \mathbf{R_p}, \qquad \mathbf{R_p} = (\overrightarrow{r_1}^p, ..., \overrightarrow{r_n}^p),$$

hvor $\mathbf{R_p}$ er en øvre trekantsmatrix. Vi kan derfor skrive k_{ij}^p som

$$k_{ij}^p = \overrightarrow{r_i}^{pT} \overrightarrow{r_j}^p = \sum_{k=1}^{n} r_{ik}^p r_{jk}^p.$$

Til sidst kan vi udtrykke $\overrightarrow{\zeta}^T \mathbf{E} \overrightarrow{\zeta}$ som

$$\begin{aligned}
\overrightarrow{\zeta}^T \mathbf{E} \overrightarrow{\zeta} &= \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\sum_{k=1}^{n} r_{ik}^1 r_{jk}^1 \right) \left(\sum_{l=1}^{n} r_{il}^2 r_{jl}^2 \right) \\
&= \sum_{k,l=1}^{n} \sum_{i,j=1}^{n} \zeta_i \zeta_j r_{ik}^1 r_{jk}^1 r_{il}^2 r_{jl}^2 \\
&= \sum_{k,l=1}^{n} \left(\sum_{i=1}^{n} \zeta_i r_{ik}^1 r_{ik}^2 \right) \left(\sum_{j=1}^{n} \zeta_j r_{jk}^1 r_{jl}^2 \right) \\
&= \sum_{k,l=1}^{n} \left(\sum_{i=1}^{n} \zeta_i r_{ik}^1 r_{il}^2 \right)^2 \geq 0.
\end{aligned}$$

Dermed er $\mathbf{E}$ positiv semi-definit, hvormed $K(\overrightarrow{x_i}, \overrightarrow{x_j})$ er en kerne.

4) $\quad K(\overrightarrow{x_i}, \overrightarrow{x_j}) = f(\overrightarrow{x_i}) f(\overrightarrow{x_j})$

Vi prikker igen $\overrightarrow{\zeta}$ på hver side af funktionen

$$\begin{aligned}\overrightarrow{\zeta}^T f(\overrightarrow{x_i}) f(\overrightarrow{x_j}) \overrightarrow{\zeta} &= \sum_{i=1}^{n}\sum_{j=1}^{n} \zeta_i \zeta_j f(x_i) f(x_j) \\ &= \sum_{i=1}^{n} \zeta_i f(x_i) \sum_{j=1}^{n} \zeta_j f(x_j) \\ &= \left(\sum_{i=1}^{n} \zeta_i f(x_i) \right)^2 \geq 0, \end{aligned} \tag{26}$$

hvormed funktionen $K(\overrightarrow{x_i}, \overrightarrow{x_j})$ er positiv semi-definit. Da faktorers orden er irrelevant, er funktionen også symmetrisk, hvormed denne er en kerne.

Vi har dermed bevist, at 1) summen af to kerner er en kerne, 2) en konstant ganget på en kerne er en kerne, 3) to kerner ganget med hinanden er en kerne og 4) to funktioner ganget med hinanden, hvor $f : X \to \mathbb{R}, X \subseteq \mathbb{R}^n$, er en kerne.

Sætter vi $f(\cdot) = c$, får vi, at $K(\overrightarrow{x_i}, \overrightarrow{x_j}) = c^2$, som vi kaldte J_2 ovenfor, også er en kerne. Dermed har vi bevist, at J_1, J_2 og J_3 er kerner og dermed som følge af 1), at disse lagt sammen også er en kerne, hvormed den polynomiske kerne af anden grad er en kerne.

Opstiller vi den polynomiske kerne af tredje grad, nemlig

$$\begin{aligned} K(\overrightarrow{x_i}, \overrightarrow{x_j}) &= (\gamma \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle + c)^3 \\ &= \gamma^3 \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle + c^3 \\ &+ 3\gamma c^2 \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle + 3\gamma^2 c \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle, \end{aligned}$$

bliver det klart, at beviset kan generaliseres til $d = n$, idet den polynomiske kerne med et vilkårligt d altid vil kunne sammensættes af de fire funktioner, vi beviste var kerner ovenfor.

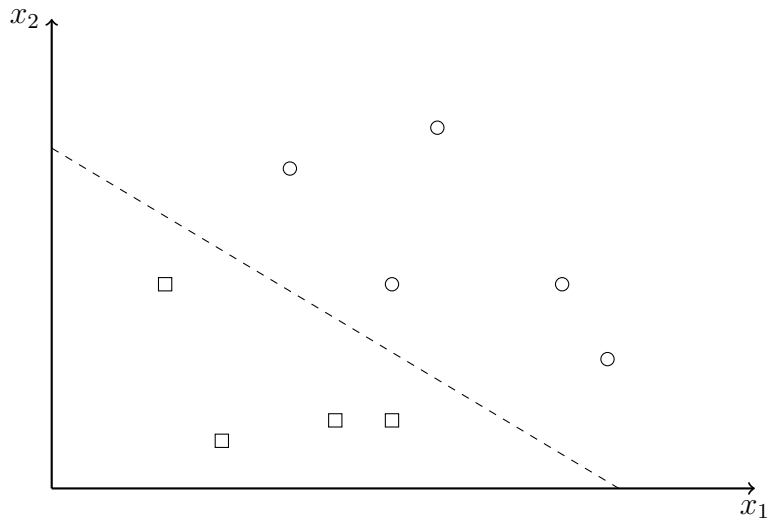
■

6.4.4.2 Den lineære kerne

Den lineære kerne er et specialtilfælde af den polynomiske kerne defineret ovenfor i ligning (24) med $d = 1$. Sætter vi $f(\cdot) = \sqrt{c}$ i 4) fra beviset ovenfor, kan vi dog se af ligning (26), at vi kan tillade, at c også må tage negative værdier. Den lineære kerne er dermed defineret ved

$$K(\overrightarrow{x_i}, \overrightarrow{x_j}) = \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle + c, \qquad c \in \mathbb{R},\ i, j = 1, ...n.$$

Når vi senere implementerer kernen i programmet R, har kernen $c = 0$, hvormed det blot bliver det indre produkt. Dette betyder, at der ikke er nogen parametre, der er specifikke for den lineære kerne ved implementeringen i R. Den lineære kerne adskiller altså data ved lineære hyperplaner i det originale inputrum. Nedenfor er illustreret en situation, hvor den lineære kerne vil være det optimale kernevalg.



Figur 13

6.4.4.3 RBF-kernen

Radial basis function kernen eller den gaussiske kerne, som vi kalder *RBF-kernen*, er en af de mest anvendte kerner i SVM, hvorfor denne også er sat som standard kerne i R ved pakken 'e1071', som vi vil benytte i analysen. RBF-kernen opererer i det

uendelig-dimensionale rum og udnytter dermed denne egenskab for et Hilbertrum. Generelt er kernen defineret som

$$K(\overrightarrow{x_i}, \overrightarrow{o_j}) = e^{-\frac{||\overrightarrow{x_i}-\overrightarrow{o_j}||^2}{2\sigma^2}}, \quad i, j = 1, ..., n, \tag{27}$$

hvor o_j er *orienteringspunkter*. Når vi implementerer RBF-kernen i R, er det kun en γ parameter, der skal vælges.

I nedenstående eksempel vælger vi orienteringspunkterne manuelt for blot at illustrere intuitionen bag kernen. I teorien er disse defineret iterativt som hver observation i træningssættet. Ved implementeringen af kernen i R er der indført et $\gamma = 1/2\sigma^2$, hvormed kernen skrives som

$$K(\overrightarrow{x_i}, \overrightarrow{o_j}) = e^{-\gamma||\overrightarrow{x_i}-\overrightarrow{o_j}||^2}, \quad i, j = 1, ..., n, \ \gamma \in \mathbb{R}^+.$$

Som nævnt før transformerer RBF-kernen ud i det uendelig-dimensionale rum. Dette kan vi indse, hvis vi eksempelvis ganger kernefunktionen ud ved $\gamma = 1/2$.

$$\exp\Big(-\frac{1}{2}||\overrightarrow{x}-\overrightarrow{o}||^2\Big) = \exp\Big(-\frac{1}{2}||\overrightarrow{x}||^2\Big)\exp\Big(-\frac{1}{2}||\overrightarrow{o}||^2\Big)\exp\Big(2\langle\overrightarrow{x}\cdot\overrightarrow{o}\rangle\Big).$$

Sidste funktion kan nu skrives op som en uendelig sum, således at vi får

$$= \exp\Big(-\frac{1}{2}||\overrightarrow{x}||^2\Big)\exp\Big(-\frac{1}{2}||\overrightarrow{o}||^2\Big)\sum_{k=0}^{\infty}\frac{\big(\langle\overrightarrow{x}\cdot\overrightarrow{o}\rangle\big)^k}{k!}.$$

For overskuelighedens skyld definerer vi nu $h(\overrightarrow{x})$ ved

$$h(\overrightarrow{x}) = \exp\Big(-\frac{1}{2}||\overrightarrow{x}||^2\Big),$$

hvormed vi kan skrive kernen, som

$$\begin{aligned}\exp\Big(-\frac{1}{2}||\overrightarrow{x}-\overrightarrow{o}||^2\Big) &= \sum_{k=0}^{\infty}\frac{h(\overrightarrow{x})\big(\langle\overrightarrow{x}\cdot\overrightarrow{o}\rangle\big)^k h(\overrightarrow{o})}{k!} \\ &= \sum_{k=0}^{\infty}\Big\langle\sqrt[k]{\frac{h(\overrightarrow{x})}{k!}}\overrightarrow{x}\cdot\sqrt[k]{\frac{h(\overrightarrow{o})}{k!}}\overrightarrow{o}\Big\rangle^k,\end{aligned} \tag{28}$$

og hvor vi kan se, at vi har indre produkter af uendeligt store polynomier, der dermed transformerer data ud i det uendelige rum 6.4.4.1.

Vi vil nu arbejde videre med definitionen fra ligning (27), idet det er denne definition, der er den mest gængse inden for den teoretiske litteratur. Vi vil nu først bevise, at RBF-kernen rent faktisk er en kerne og dernæst illustrere, hvordan denne fungerer ved eksempler.

Bevis

Vi starter med at skrive kernen op på formen

$$K(\overrightarrow{x}, \overrightarrow{o}) = e^{-\frac{||\overrightarrow{x}||^2}{2\sigma^2}} e^{-\frac{||\overrightarrow{o}||^2}{2\sigma^2}} e^{-\frac{2\langle \overrightarrow{x} \cdot \overrightarrow{o} \rangle}{2\sigma^2}}.$$

Vi viste i afsnit 6.4.4.1, at en funktion af $\overrightarrow{x}$, $f : X \to \mathbb{R}, X \subseteq \mathbb{R}^n$, er en kerne, hvormed de første to faktorer ovenfor er kerner. Vi viste desuden, at to kerner ganget sammen også er en kerne. Vi mangler derfor at vise, at $\exp(2\langle \overrightarrow{x} \cdot \overrightarrow{o} \rangle / 2\sigma^2)$ er en kerne.

Vi ser først og fremmest, at $2\langle \overrightarrow{x} \cdot \overrightarrow{o} \rangle / 2\sigma^2$ er en kerne, da dette udtryk blot er en konstant ganget på identitetskernen, som vi i afsnit 6.4.4.1 viste var en kerne.

Det er et kendt resultat, at enhver eksponentialfunktion kan approksimeres med et polynomium. Eksempelvis kan $g(x) = e^x$ approksimeres ved $h(x) = (1 + x/n)^n$, for n gående mod uendelig. Da vi allerede har vist, at den polynomiske kerne er en kerne, må eksponentialfunktionen til denne kerne $\exp(K(\overrightarrow{x}, \overrightarrow{o}))$ også være en kerne.

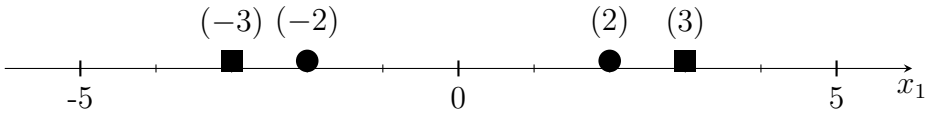
■

Vi vil nu vise ved et eksempel, hvordan kernen bruges til at adskille observationer lineært, der ikke er lineært separable i deres originale rum, samt hvordan beregninger ikke foretages i det projicerede rum men i det originale inputrum.

Eksempel 4.

Anvender vi observationerne fra eksempel 1, kan vi bruge kernetricket til at undgå at udregne transformationerne og dermed foretage beregninger i et rum af højere dimension. Vi havde observationerne

$$S = (\overrightarrow{x}_1, \overrightarrow{x}_2, \overrightarrow{x}_3, \overrightarrow{x}_4) = \{(2), (-2), (3), (-3)\}.$$



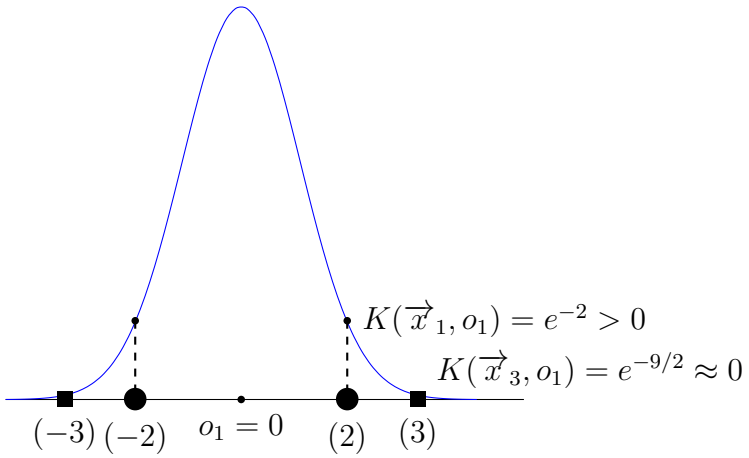
Figur 14

Vi sætter her manuelt orienteringspunktet $o_1 = 0$ og sætter $\sigma = 1$, så

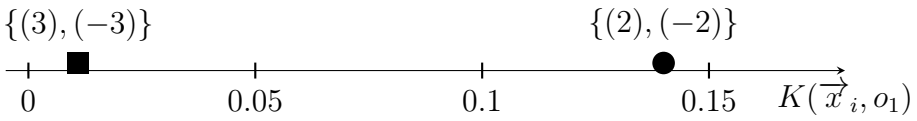
$$K(\overrightarrow{x}_1, o_1) = e^{-\frac{||\overrightarrow{x}_1 - o_1||^2}{2\sigma^2}} = e^{-\frac{||2-0||^2}{2 \cdot 1^2}} = e^{-2}.$$

Ved samme formel får vi

$$K(\overrightarrow{x}_2, o_1) = e^{-2}, \qquad K(\overrightarrow{x}_3, o_1) = e^{-9/2}, \qquad K(\overrightarrow{x}_4, o_1) = e^{-9/2}$$



Figur 15



Figur 16

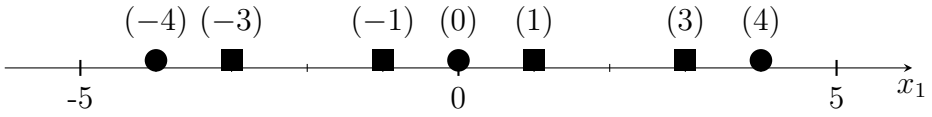
Af figur 15 og 16 ses det, at jo tættere observationen er på orienteringspunktet o_1, des større er værdien af kernefunktionen $K(\overrightarrow{x}_i, o_1)$. Mere specifikt bruger vi altså kernen til at klassificere ud fra om $K(\overrightarrow{x}_i, o_1) > 0$ eller om $K(\overrightarrow{x}_i, o_1) \approx 0$.

Vi bemærker samtidig, at kernen udregnes i det endimensionale rum og dermed ikke bruger regnekraft på at projicere observationerne ind i et rum, som vi ved har den uendelige dimension fra ligning (28). Desuden bemærker vi, at σ er den eneste parameter, vi kan ændre på i RBF-kernen. Vi skal i næste eksempel se på, hvordan σ ændrer vores klassifikation (*Ng, 2018*).

□

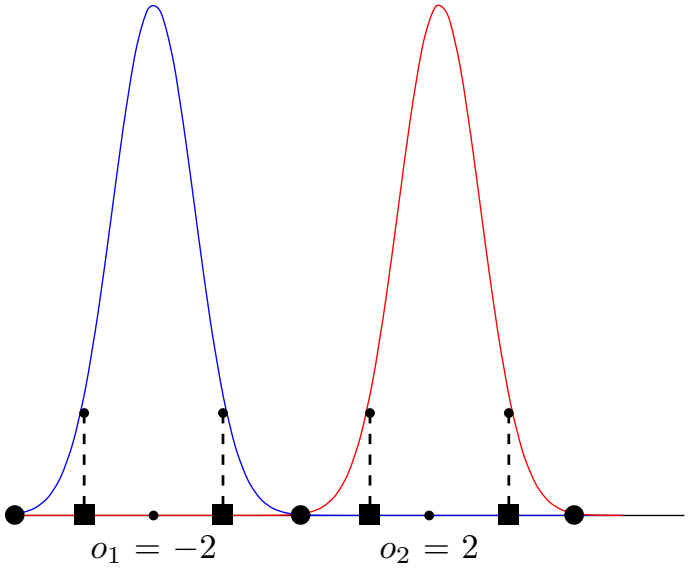
Eksempel 5.

Betragt følgende datasæt.
Klasse 1: $(\overrightarrow{x}_1, \overrightarrow{x}_2, \overrightarrow{x}_3, \overrightarrow{x}_4) = \{(-3), (-1), (1), (3)\}$
Klasse 2: $(\overrightarrow{x}_5, \overrightarrow{x}_6, \overrightarrow{x}_7) = \{(-4), (0), (4)\}$



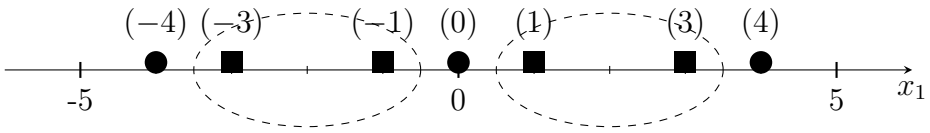
Figur 17

Anvendes metoden fra forrige eksempel med ét orienteringspunkt, er det ikke muligt at adskille de to klasser, men ved at indføre to orienteringspunkter, $o_1 = -2$ og $o_2 = 2$, bliver det muligt.



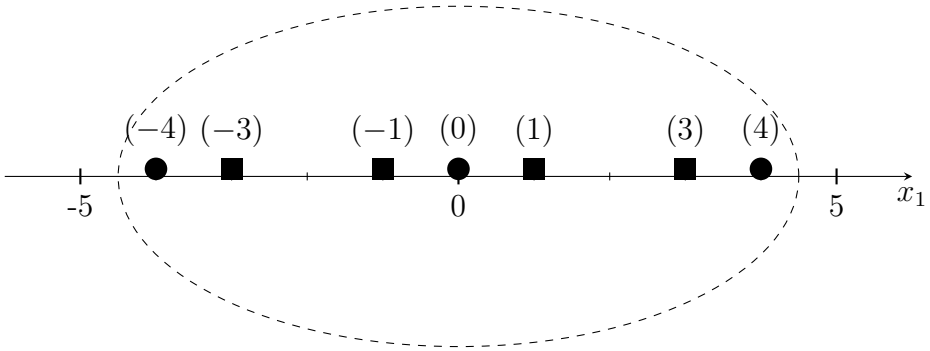
Figur 18

I forhold til forrige eksempel klassificerer vi nu ud fra om $K(\overrightarrow{x}_i, o_1) + K(\overrightarrow{x}_i, o_2) > 0$ eller $K(\overrightarrow{x}_i, o_1) + K(\overrightarrow{x}_i, o_2) \approx 0$. Ved den nuværende værdi af σ adskiller kernen altså punkterne korrekt, hvilket kan illustreres som på figuren herunder i det oprindelige rum.



Figur 19

Gør vi σ større, vil vores gaussfunktioner i figur 18 flade ud. I det originale rum fra overstående figur betyder det, at de separerende cirkler bliver større. Dette giver en mere glat model men med flere misklassifikationer som illustreret på næste figur, hvor vi får tre misklassificerede observationer.



Figur 20

Ved blot at justere på σ kan vi altså ud fra klassifikationsraterne på trænings- og testsættet finde den optimale klassifikationsmodel, der klassificerer godt på træningssættet og samtidig kan generaliseres over på testsættet.

□

6.4.4.4 Valg af kerne

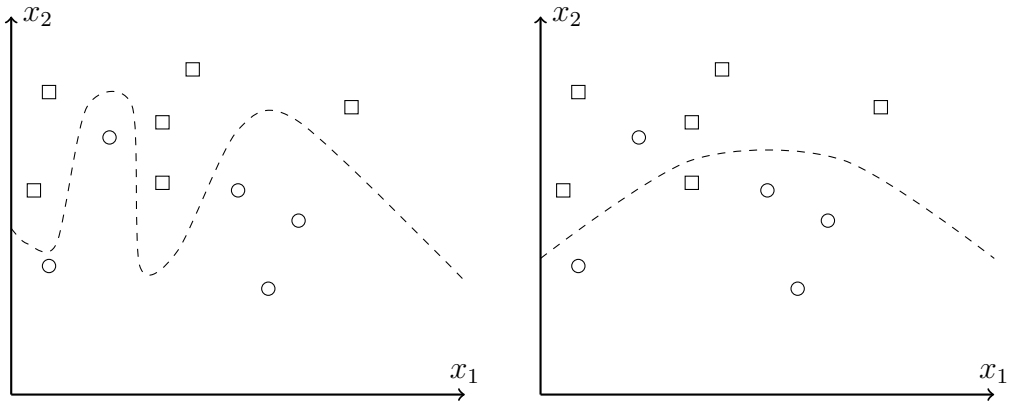
Det at vælge den bedste kerne afhænger som sagt af det konkrete datasæt. Arbejder man med et en-, to- eller tredimensionalt datasæt, er det muligt at sige noget kvalificeret om, hvilken kerne der vil passe bedst til situationen. Hvis data kan adskilles med lineære hyperplaner i det originale inputrum, vil den lineære kerne give det bedste resultat som vist på figur 13. Kan data derimod adskilles ved en n-dimensional kugle, er det RBF-kernen, man skal benytte jf. eksempel 5 og 4. Som vist i figur 12 er det den polynomiske kerne, der skal vælges, hvis data kan adskilles af et n'te grads polynomium.

I praksis vil man dog sjældent have en-, to- eller tre-dimensionalt data, hvormed det bliver svært at danne sig et overblik over hvilken kerne, der passer bedst til situationen. Valget af kerne bliver derfor i praksis ofte foretaget ved at prøve sig frem, indtil den bedste model er fundet.

Efter introduktionen af kerner, hvor vi har mulighed for at arbejde i et rum med en højere dimension, er der større risiko for, at vores model kan være svær at generalisere fra vores træningssæt til vores testsæt. Dette problem vil vi beskrive nærmere i næste afsnit.

6.5 Generaliseringsteori

Generaliseringsteori er en vigtig teori for machine learning og specielt ved Supervised Learning. Brugen af generaliseringsteori sikrer, at den model, der bliver bygget, ikke kun kan bruges til vores specifikke datasæt, men at den også kan bruges mere generel,t og dermed vil kunne klassificere observationer i andre lignende datasæt. Grunden til at en model, der beskriver vores data perfekt, ikke nødvendigvis passer godt på mere generelle tilfælde er, at vores stikprøve ikke er komplet og ofte vil indeholde støj. Senere vil vi beskrive, hvordan man kan løse eller i hvert fald mindske dette problem, men først vil vi forklare, hvilke problemer det skaber.



Figur 21

I figur 21 kan vi se, at modellen på venstre side gør alt for at klassificere alle observationer korrekt. Ofte er det en bedre idé at lave en mere generel model, som vi kan se på højre side af figuren, der misklassificerer enkelte observationer, da denne til gengæld sandsynligvis bedre kan generalisere og bruges på ny lignende data. Dermed

er graden af, hvor godt modellen fitter træningssættet, ikke direkte et mål for hvor god modellen er.

Problemet med modellen til venstre i figur 21 kaldes *overfitting*. Overfitting er et essentielt problem især efter introduktionen af kerner, der tillader os at bevæge os op i et rum med en højere dimension.

Overfitting kan forekomme, når det forsøges at prædiktere et datasæt med meget støj, eller når der forsøges med en kompliceret kerne til et simpelt klassifikationsproblem. Problemet med overfitting kan observeres ved, at man har en model med en høj klassifikationsrate på træningssættet, men man får en relativ lav klassifikationsrate, når man tester modellen på testsættet. Vi forsøger altså, at finde en model der er robust, dvs. en model der har stort set lige stor klassifikationrate på trænings- og testsættet. En model er dog ikke god, bare fordi den er robust. Et eksempel på dette kunne være *underfitting*.

Underfitting defineres som tilfældet, hvor modellen hverken er en god model for vores træningssæt eller for vores testsæt. Dette problem er dog meget nemt at opdage, da vores model vil have en lav klassifikationsrate for både trænings- og testsættet.

Problemet med underfitting kan forekomme, når modellen har svært ved, at finde en sammenhæng mellem de beskrivende variable og klassen som vi ønsker at klassificere. En underfittet model kan sagtens være robust, men hvis klassifikationsraten er for lav, vil det ikke være en model, vi kan bruge.

Vi kan undgå generaliseringsproblemerne på flere forskellige måder. Den mest simple måde er at indsætte mere data. Det er dog ikke altid, at det er muligt, eller at det hjælper. Ved at bruge mere data kan vi i mange tilfælde mindske den indflydelse, som støj i et datasæt har på den endelige model. Dog hjælper det ikke, hvis det data, man tilføjer, også er fyldt med støj.

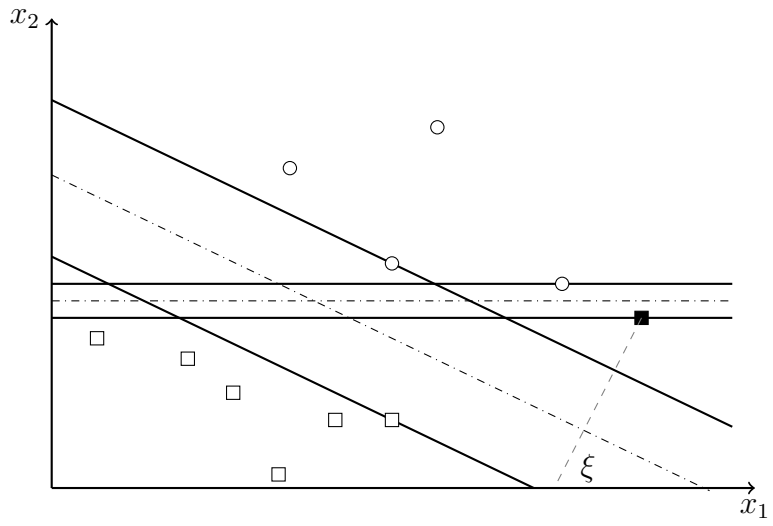
Andre måder, hvorpå vi kan undgå overfitting, er ved at benytte en metode, der kaldes for krydsvalidering, som er beskrevet i afsnit 8.3.1, eller ved at vælge en kerne som er bedre til at generalisere på det specifikke datasæt, samtidig med at vi har en høj klassifikationsrate.

Der findes flere metoder, hvorpå vi kan undgå overfitting, men de bedste og mest anvendte er som nævnt ovenfor, og disse er også dem, som vi vil benytte os af i analysen.

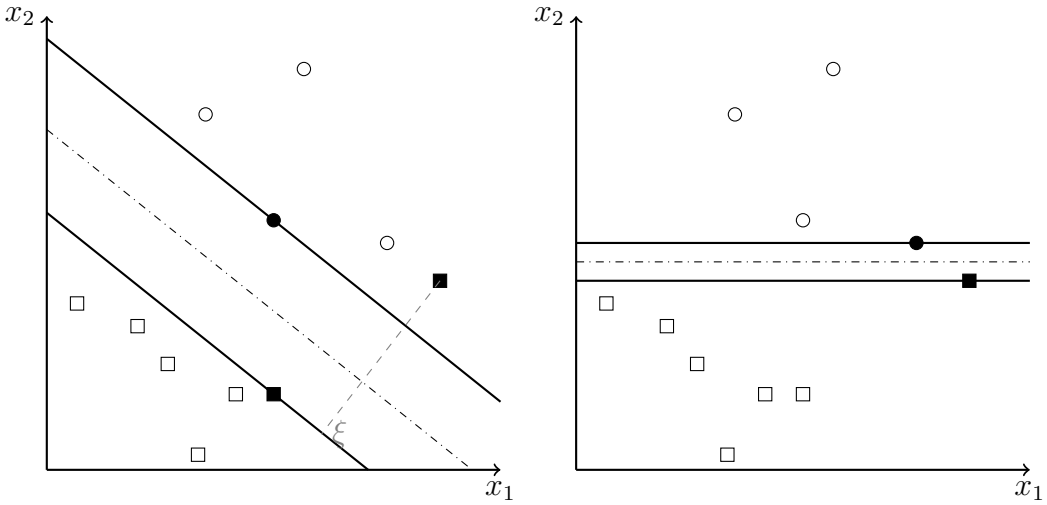
6.5.1 Outliere

En af de ting, der kan medføre problemer med generaliseringen, er outliere. Hvis data indeholder mange outliere, kan det føre til en lang række af misklassifikationer.

På figur 22 kan vi se, hvordan et separerende hyperplan ville se ud med blød margin og hård margin. Her er den fyldte firkant en outlier. Vi kan altså se, at den hårde margin er meget sensitiv i forhold til outliere, men da vi har introduceret ideen om blød margin, der tillader misklassifikationer, gør dette, at outliere ikke får så stor indflydelse på marginen, som i tilfældet ved hård margin. Dette betyder, at SVM er mere robust overfor outliere efter introduktionen af den bløde margin og slackvariablen. Vi kan desuden videre udlede, at en højere C leder til at outliere, der også er misklassificerede, har større effekt, og hyperplanet bliver derfor mindre robust over netop outliere. Dette skyldes, at $C \to \infty$ gør marginen hårdere jf. afsnit 6.3.2.2. Dette er illustreret på figur 22 og 23.



Figur 22



Figur 23

I figur 23 har vi en lav C i venstre graf og en høj C i højre graf, hvor det med stor sandsynlighed er modellen på venstresiden, der er bedst til at generalisere over på et andet datasæt. Herudfra kan vi udlede, at en blød margin vil være bedre til at håndtere data med støj ved tilladelsen af misklassifikationer, hvormed modellen sandsynligvis bedre kan generalisere på ny data.

6.6 Normalisering

Når der i Machine Learining indlæses data, er det meget sjældent, at alle de forklarende variable arbejder i samme størrelsesorden. Hvis vi eksempelvis ønsker, at klassificere hvorvidt en bil er en sportsbil eller ej, kunne man have forklarende variable som pris, topfart og antal sæder. Alle tre variable arbejder i vidt forskellige intervaller, hvilket vil betyde, at værdier, der arbejder i meget større intervaller, vil have langt større indflydelse på det endelige resultat. Vi kan undgå dette ved at normalisere data.

Der findes flere forskellige måder at normalisere data på, alt efter hvilken fordeling data har. De to mest almindelige former for normalisering er *min/max-normalisering* og *standardisering*.

Min/max-normalisering

$$\tilde{X} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Standardisering

$$\tilde{X} = \frac{X - \mu}{\sigma}$$

Min/max-normalisering skalerer alle observationer til at være i intervallet [0;1]. Standardisering giver i stedet en middelværdi på 0 og en varians på 1, hvilket gør, at de fleste observationer ligger i intervallet [-1;1] men med mulighed for både større og mindre værdier. Som regel er standardisering foretrukket, eftersom min/max-normalisering kun skaleres efter to værdier, nemlig den største og den mindste værdi. Dette medfører, at normaliseringen vil være påvirket af støj.

7 Support Vector Machine

Support Vector Machine er en metode til at finde et optimalt separerende hyperplan ved at gøre brug af resultaterne fundet i optimeringsteori, lineær klassifikationsteori, kerneteori og generaliseringsteori.

7.1 SVM ved hård margin

Vi vil her vise, hvordan SVM gør brug af de forskellige teorier til at klassificere observationer. Vi starter med at anvende den hårde margin, og antager dermed at observationerne er lineært adskilligt. Vi vil vise, hvordan vi kan bruge den duale repræsentation af optimeringsproblemet til at gøre det muligt at anvende kerner i objektfunktionen. Ved brug af optimeringsteori vil vi sikre, at løsningen er optimal. Dernæst vil vi vise, at vi kan skrive beslutningsfunktionen udelukkende ved supportvektorer. Til sidst vil vi vise, at vi kan simplificere udtrykket for marginen.

Det primale optimeringsproblem

Vi ved fra teorien om lineær klassifikation afsnit 6.3, at et mål for det bedst separerende hyperplan er det med den maksimale geometriske margin. Vi antager i dette afsnit, at data $S = ((\overrightarrow{x_1}, y_1), ..., \overrightarrow{x_s}, y_s))$ er lineært separabelt. Det primale optimeringsproblem kan dermed skrives som

$$\begin{aligned} \min \quad & \tfrac{1}{2}||\overrightarrow{w}||^2 \\ \text{ubb.} \quad & y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) - 1 \geq 0 \end{aligned}$$

for $i = 1, ..., s$ og med geometrisk margin $M = 1/||\overrightarrow{w}||$.

For at finde den optimale løsning til dette kvadratiske optimeringsproblem kan vi opstille Karush-Kuhn-Tucker betingelserne fra afsnit 6.2.4 ligning (1) til (5). Vi opstiller derfor først den tilhørende Lagrangefunktion,

$$L(\overrightarrow{w}, b, \overrightarrow{\alpha}) = \frac{1}{2}\langle \overrightarrow{w} \cdot \overrightarrow{w} \rangle - \sum_{i=1}^{s} \alpha_i [y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) - 1],$$

hvor Lagrangemultiplikatoren $\alpha_i \geq 0$. Vi skal nu differentiere denne med hensyn til de primale variable $\overrightarrow{w}$ og b, dvs.

$$\frac{\partial L(\overrightarrow{w}, b, \overrightarrow{\alpha})}{\partial \overrightarrow{w}} = \overrightarrow{w} - \sum_{i=1}^{s} y_i \alpha_i \overrightarrow{x_i} = \overrightarrow{0} \tag{29}$$

$$\frac{\partial L(\overrightarrow{w}, b, \overrightarrow{\alpha})}{\partial b} = -\sum_{i=1}^{s} y_i \alpha_i = 0. \tag{30}$$

Vi mangler nu at opstille kravet fra ligning (3), som ved vores optimeringsproblem bliver

$$\alpha_i^* [y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x_i} \rangle + b^*) - 1] = 0, \quad i = 1, ..., s. \tag{31}$$

Det skal nu vise sig, at vi med fordel kan opstille det tilsvarende duale problem. Ved omskrivningen af ligningerne (29) og (30), får vi

$$\overrightarrow{w} = \sum_{i=1}^{s} y_i \alpha_i \overrightarrow{x}_i \tag{32}$$

$$0 = \sum_{i=1}^{s} \alpha_i y_i. \tag{33}$$

Disse kan substitueres ind i Lagrangefunktionen fra det primale optimeringsproblem og giver således den duale repræsentation som funktion af $\overrightarrow{\alpha}$

$$\begin{aligned}
L(\overrightarrow{w}, b, \overrightarrow{\alpha}) &= \frac{1}{2}\langle \overrightarrow{w} \cdot \overrightarrow{w} \rangle - \sum_{i=1}^{s} \alpha_i [y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) - 1] \\
&= \frac{1}{2} \sum_{i,j=1}^{s} \alpha_i \alpha_j y_i y_j \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle - \sum_{i=1}^{s} \alpha_i y_i \left(\overrightarrow{x}_i \left(\sum_{j=1}^{s} \alpha_j y_j \overrightarrow{x}_j \right) + b \right) + \sum_{i=1}^{s} \alpha_i \\
&= \frac{1}{2} \sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle \\
&- \sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle - \sum_{i=1}^{s} \alpha_i y_i b + \sum_{i=1}^{s} \alpha_i. \\
&= \sum_{i=1}^{s} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle - 0 \cdot b \\
&= \sum_{i=1}^{s} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle,
\end{aligned} \tag{34}$$

som vi kalder den duale objektfunktion $W(\overrightarrow{\alpha})$.

Vi ser nu, at vores bibetingelse fra det primale problem svarer til g_i fra den stærke dualitetssætning i afsnit 6.2. Da denne funktion er affin, betyder det ifølge sætningen, at vi kan løse det duale maksimeringsproblem, hvor værdien af W^* vil være lig den optimale værdi for objektfunktionen i det primale optimeringsproblem L^*. Med bibetingelsen fra ligning (33) og det at Lagrangemultiplikatorer per definition skal være ikke-negative, får vi følgende maksimeringsproblem.

$$\begin{aligned} \max \quad & W(\overrightarrow{\alpha}) = \sum_{i,j=1}^{s} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle \\ \text{ubb.} \quad & \sum_{i=1}^{s} y_i \alpha_i = 0 \\ & \alpha_i \geq 0, \qquad i = 1, ..., s. \end{aligned}$$

Bestemmelse af b

Da b ikke indgår i det duale optimeringsproblem, skal denne findes ved at bruge bibetingelsen fra det primale optimeringsproblem,

$$y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) \geq 1. \qquad i = 1, .., s.$$

Fra afsnit 6.3.2 ved vi, at der altid vil være mindst to observationer, hvor denne bibetingelse vil være aktiv, nemlig for de observationer der ligger på marginen. En observation, der ligger på marginen og tilhører den negative klasse $y_i = -1$, betegner vi som i afsnit 6.3.2 ved $\overrightarrow{x}^m_-$, som skal opfylde ligning (16)

$$\langle \overrightarrow{w} \cdot \overrightarrow{x}^m_- \rangle = -1 - b$$

Og for en observation i den positive klasse, der ligger på marginen, $\overrightarrow{x}^m_+$, skal ligning (17) være opfyldt, dvs.

$$\langle \overrightarrow{w} \cdot \overrightarrow{x}^m_+ \rangle = 1 - b.$$

Vi kan dermed finde b ved

$$b^* = -\frac{\langle \overrightarrow{w}^* \cdot \overrightarrow{x}^m_- \rangle + \langle \overrightarrow{w}^* \cdot \overrightarrow{x}^m_+ \rangle}{2} = 1 - \langle \overrightarrow{w}^* \cdot \overrightarrow{x}^m_+ \rangle = -\langle \overrightarrow{w}^* \cdot \overrightarrow{x}^m_- \rangle - 1.$$

Matematisk kan observationerne på marginen findes ved at finde de observationer, hvor $y_i(\langle \overrightarrow{w}^* \cdot \overrightarrow{x}_i \rangle + b)$ er tættest på 1 og dermed størst. Med b fast for alle observationer svarer dette til at finde observationerne, der maksimerer $y_i(\langle \overrightarrow{w}^* \cdot \overrightarrow{x}_i \rangle)$. Vi kan derfor finde b ved

$$b^* = -\frac{\max_{y_i=-1}\langle \overrightarrow{w}^* \cdot \overrightarrow{x}_i\rangle + \min_{y_i=1}\langle \overrightarrow{w}^* \cdot \overrightarrow{x}_i\rangle}{2}$$
$$= 1 - \min_{y_i=1}\langle \overrightarrow{w}^* \cdot \overrightarrow{x}_i\rangle = -\max_{y_i=-1}\langle \overrightarrow{w}^* \cdot \overrightarrow{x}_i\rangle - 1. \tag{35}$$

Beslutningsfunktionen defineret ved supportvektorer

Vi vil nu vise, at vi kan skrive beslutningsfunktionen udelukkende ved supportvektorerne. I afsnit 6.3, ligning (6), definerede vi beslutningsfunktionen som

$$f(\overrightarrow{x}) = \sum_{i=1}^{s} w_i x_i + b.$$

Indsætter vi nu ligning (32), kan vi dermed skrive beslutningsfunktionen som

$$f(\overrightarrow{x}, \overrightarrow{\alpha}^*, b^*) = \sum_{i=1}^{s} y_i \alpha_i^* \langle \overrightarrow{x_i} \cdot \overrightarrow{x}\rangle + b^*. \tag{36}$$

I ligning (31) opstillede vi Karush-Kuhn-Tucker komplementaritetsbetingelsen for vores problem, som så således ud.

$$\alpha_i^*[y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x_i}\rangle + b^*) - 1] = 0, \quad i = 1, ..., s.$$

Dette medfører, at α_i^* kun kan være forskellige fra nul, hvis $y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x_i}\rangle + b^*) = 1$. Dette er kun tilfældet, når observationen ligger på marginen, som vist i afsnit 6.3.2. For alle andre observationer er $\alpha_i^* = 0$.

I løsningen af optimeringsproblemet ved hård margin er det altså kun observationerne på marginen, der bestemmer hyperplanet, og vi kalder disse observationer for supportvektorer og skriver $i \in sv$, når $\overrightarrow{x}_i$ er en supportvektor. Vi kan derfor skrive beslutningsfunktionen udelukkende ved supportvektorer

$$f(\overrightarrow{x}, \overrightarrow{\alpha}^*, b^*) = \sum_{i \in sv} y_i \alpha_i^* \langle \overrightarrow{x_i} \cdot \overrightarrow{x}\rangle + b^*.$$

Marginen

Vi vil her vise, hvordan vi kan udregne den geometriske margin M direkte ved vores

$\overrightarrow{\alpha}^*$. Hvis observation $\overrightarrow{x}_i$ er en supportvektor, ved vi, at beslutningsfunktionen af denne observation giver præcis 1 eller -1, dvs. $f(\overrightarrow{x}_i) = \{-1, 1\},\ \ i \in sv$, og vi kan derfor skrive

$$y_j f(\overrightarrow{x}_j, \overrightarrow{\alpha}^*, b^*) = y_j\big(\sum_{i=1}^{s} y_i \alpha_i^* \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle + b^*\big) = 1 \tag{37}$$

$$\Rightarrow \sum_{i \in sv} y_i \alpha_i^* \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle = 1/y_j - b^*$$

Bruger vi overstående resultat sammen med ligning (32) og (33), og det at $\alpha_i = 0,\ i \notin sv$, så kan vi skrive

$$\langle \overrightarrow{w}^* \cdot \overrightarrow{w}^* \rangle = \sum_{i,j=1}^{s} y_i y_j \alpha_i^* \alpha_j^* \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle \tag{38}$$

$$= \sum_{j \in sv} \alpha_j^* y_j \sum_{i \in sv} y_i \alpha_i^* \langle \overrightarrow{x_i} \cdot \overrightarrow{x_j} \rangle = \sum_{j \in sv} \alpha_j^* y_j (1/y_j - b^*) \tag{39}$$

$$= \sum_{j \in sv} \alpha_j^* (1 - y_j b^*) = \sum_{j \in sv} \alpha_j^* - y_j \alpha_j^* b^* \sum_{j \in sv} \alpha_j^* - 0 \cdot b^* \tag{40}$$

$$= \sum_{j \in sv} \alpha_j^*. \tag{41}$$

Vi ved, at en vektor prikket med sig selv giver vektorens kvadrerede længde. Dermed kan vi skrive marginen som

$$M = 1/||\overrightarrow{w}|| = \frac{1}{\sqrt{\langle \overrightarrow{w}^* \cdot \overrightarrow{w}^* \rangle}} = \left(\sum_{i \in sv} \alpha_i^* \right)^{-1/2}.$$

Kerner i objektfunktionen

Vi er nu kommet frem til, at vi kan skrive det primale optimeringsproblem om til et kvadratisk optimeringsproblem ved hjælp af den duale repræsentation, hvori data kun indgår som indre produkter. Derfor kan vi også anvende kernetricket fra afsnit 6.4.1 og indsætte en vilkårlig kerne i den duale objektfunktion. Dermed får vi

$$W(\overrightarrow{\alpha}) = \sum_{i=1}^{s} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j \langle \overrightarrow{\phi}(\overrightarrow{x_i}) \cdot \overrightarrow{\phi}(\overrightarrow{x_j}) \rangle$$

eller

$$W(\overrightarrow{\alpha}) = \sum_{i=1}^{s} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j K(\overrightarrow{x_i}, \overrightarrow{x_j}). \tag{42}$$

Det smarte ved kernetricket er, at vi ikke behøver kende hele $\overrightarrow{\phi}(\overrightarrow{x})$ men kun de indre produkter mellem $\overrightarrow{\phi}(\overrightarrow{x_i})$ og $\overrightarrow{\phi}(\overrightarrow{x_j})$. Dette sparer os for meget beregningskraft, især når $\overrightarrow{\phi}$ arbejder i et rum af en høj dimension. Vi kan nemt se, at det er sandt i den duale objektfunktion, men vi bliver nødt til at vise, at det også holder i bibetingelserne. Hvis vi kigger på de første bibetingelser

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

og

$$\alpha_i \geq 0,$$

kan vi se at disse ikke afhænger af $\overrightarrow{x}$. Derfor er det ikke nødvendigt for os, at vide hvad $\overrightarrow{\phi}$ er.

Vi kigger nu i stedet på vores beslutningsfunktion

$$f(\overrightarrow{x}) = \sum_{i=1}^{m} w_i x_i + b \Rightarrow \sum_{i=1}^{m} w_i \overrightarrow{\phi}(x_i) + b,$$

som vi viste i ligning (36) kunne skrives som et indre produkt, så

$$\begin{aligned} f(\overrightarrow{x}) &= \sum_{i=1}^{s} \alpha_i y_i \langle \overrightarrow{\phi}(\overrightarrow{x_i}) \cdot \overrightarrow{\phi}(\overrightarrow{x}) \rangle + b \\ &= \sum_{i=1}^{s} \alpha_i y_i K(\overrightarrow{x_i}, \overrightarrow{x}) + b. \end{aligned}$$

Her har vi igen kun brug for at kende et indre produkt mellem $\overrightarrow{\phi}$'erne. Vi har nu vist, at det ikke er nødvendigt at kende til $\overrightarrow{\phi}$ for at bruge SVM. Dette betyder, at vi ikke direkte foretager transformationen af data ind i et andet rum, der potentielt set kan være et uendelig-dimensionalt rum. Dermed sparer vi den ekstra beregningskraft, det ville have krævet at beregne transformationen af vores data eksplicit som illustreret i eksempel 5.

Konveksitet

Vi vil nu vise, at den duale objektfunktion fra ligning (42) er konveks, således at løsningen altid er optimal. For at dette gælder skal sidste led altså være konveks, idet dette led har negativt fortegn.

$$W(\overrightarrow{\alpha}) = \sum_{i=1}^{s} \alpha_i - \frac{1}{2} \sum_{i=1}^{s} \sum_{j=1}^{s} y_i y_j \alpha_i \alpha_j K(\overrightarrow{x_i}, \overrightarrow{x_j})$$
$$= \overrightarrow{\alpha}^T \cdot \overrightarrow{1} - \frac{1}{2}(\overrightarrow{\alpha}^T \cdot (\text{diag}(\mathbf{Y}) \cdot \mathbf{K} \cdot \text{diag}(\mathbf{Y})) \cdot \overrightarrow{\alpha}),$$

hvor

$$\mathbf{Y} = \begin{bmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & y_n \end{bmatrix}$$

og hvor $\mathbf{K}$ er den samme matrix som beskrevet i afsnit 6.4.2.

Måden, hvorpå vi afgør om sidste er konvekst, er ved at undersøge, hvorvidt $\mathbf{K}$ er en positiv semidefinit matrice. Lige netop dette beviste vi i afsnit 6.4.2. Vi kan derfor konkludere, at vores duale objektfunktion er konveks.

Det duale optimeringsproblem

Samler vi disse fire resultater, kan vi skrive vores primale optimeringsproblem om til det samlede duale optimeringsproblem.

$$\begin{aligned} max \qquad & W(\overrightarrow{\alpha}) = \sum_{i=1}^{s} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j K(\overrightarrow{x_i}, \overrightarrow{x_j}) \\ u.b. \qquad & \sum_{i=1}^{s} y_i \alpha_i = 0 \\ & \alpha_i \geq 0, \end{aligned}$$

for $i = 1, ..., s$, med beslutningsfunktionen $f(\overrightarrow{x}) = \sum_{i \in sv} y_i \alpha_i^* K(\overrightarrow{x_i}, \overrightarrow{x}) + b^*$, hvormed vi kan klassificere observationer ved beslutningsreglen $\text{sgn}(f(\overrightarrow{x}))$, og med geometrisk margin $M = \left(\sum_{i \in sv} \alpha_i^* \right)^{-1/2}$.

7.2 SVM ved blød margin

Vi vil nu vise, hvordan SVM gør brug af teorien om blød margin til at klassificere observationer i klasser. Vi vil i dette afsnit støde på 1-norm og 2-norm blød margin, som definerer, hvordan observationer med slack skal straffes. Vi vil som i afsnittet om SVM ved hård margin også opstille det primale og udlede det duale problem. Desuden vil vi opstille udtrykket for den optimale margin og bestemme b ved blød margin.

Som nævnt i afsnit 6.3.2.2 om den bløde margin er det ofte ikke muligt at separere observationer lineært. I afsnit 6.4 omhandlende kerner nævnte vi, at det i teorien altid er muligt at finde en kerne, der gør det muligt at separere data lineært. Dette er dog ikke altid hensigtsmæssigt. For det første fordi at det kan kræve lang tid at finde den korrekte kerne, og for det andet fordi det potentielt kan lede til overfitting. Vi vil derfor inkorporere slackvariablen i vores optimeringsmodel.

Ofte følger data ikke en model uden støj, og der vil derfor ofte være outliere, som vi gerne vil undgå kommer til at have for meget indflydelse på vores klassifikationsmodel. I vores datasæt kan en outlier eksempelvis skyldes en usædvanlig begivenhed eller fejl i data. At tillade en slackvariabel giver derfor ofte god mening, hvilket også er tilfældet ved vores datasæt.

Vi introducerede slackvariablen i afsnittet 6.3.2.2 om bløde marginer. Når vi bruger SVM med bløde marginer, har vi følgende optimeringsproblem.

$$\begin{aligned} \min \quad & \frac{1}{2}||\overrightarrow{w}||^2 + C\sum_{i=1}^{s}\xi_i^k \\ \text{ubb.} \quad & y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \end{aligned} \tag{43}$$

for $i = 1, ..., s$, og hvor $k = \{1, 2\}$, som definerer hvorvidt problemet er blød margin ved 2-norm eller 1-norm.

7.2.1 Forskellen på 1-norm og 2-norm

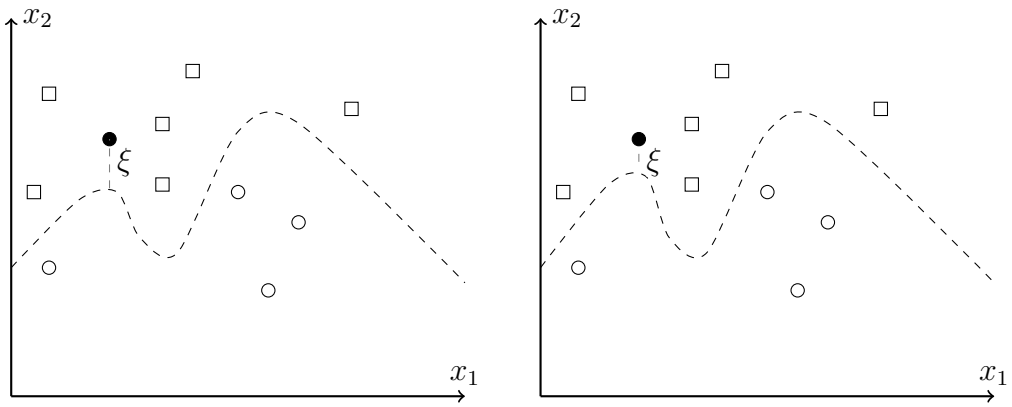
Ved 1-norm straffes observationernes slack hårdere, når værdien af C øges, fordi C ganges på summen af slackvariablerne. Dermed vil en forøgelse af C have den samme relative forøgelse af straf for al slack.

Ser vi på objektfunktionen ligning (43) fra optimeringsproblemet, når misklassificerede observationer tillades, med hinge-loss funktionen fra (20)

$$\xi_i = \max\left(0, 1 - y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b)\right), \qquad = 1, ..., s$$

ser vi, at vi har mulighed for at gøre vores margin og dermed også det separerende hyperplan endnu mere sensitivt overfor observationer, der ligger relativt langt væk fra marginen. Af definitionen på slack ovenfor ses det, at observationer under 1-norm blot straffes med deres afstand ind til marginen ganget med C. Det skal vise sig, at en given observations indflydelse på det separerende hyperplan er begrænset under 1-norm, idet $0 \leq \alpha_i \leq C$.

Ved at justere på k har vi mulighed for at øge forskellen på den straf, som vi tillægger misklassificerede observationer tæt på marginen, i forhold til de misklassificerede observationer der ligger relativt langt fra marginen. Ved at øge k fra $k = 1$ til $k = 2$ straffer vi misklassificerede observationers med deres kvadrerede afstand ind til marginen ganget med C.



Figur 24

Figuren ovenfor illustrerer, hvordan den misklassificerede observation, markeret med sort udfyldning, har større indflydelse på det separerende hyperplan under 2-norm (højre) sammenlignet med 1-norm (venstre), selvom værdien af C er den samme. Ved optimeringsproblemet med 2-norm blød margin får den misklassificerede observation nemlig vægtet slacken højere i objektfunktionen, som der skal minimeres, hvormed hyperplanet tilpasses observationen relativt mere.

I figuren har vi anvendt den polynomiske kerne, hvormed hyperplanet er et polynomium, når det projiceres tilbage i det originale rum. Forskellen er altså, at vi i 2-norm får et mindre glat polynomium, der dog ligger tættere på den misklassificerede observation i forhold til 1-norm.

Om man skal vælge 1-norm eller 2-norm afhænger af, hvilken fordeling data følger. Indeholder data ekstreme observationer, vil 2-norm oftere klassificere disse korrekt, da der vil foretages færre misklassifikationer, der hvor observationerne ligger tæt på hinanden. Er disse ekstreme observationer derimod blot støj, så vil 1-norm modelleringen være den mest robuste overfor disse.

7.2.2 1-norm blød margin

Optimeringsproblemet med blød margin, hvor vi antager at $k = 1$, kommer fra teorien om den bløde margin, som vi kender fra afsnit 6.3.2.2. Vi vil nu vise, hvordan SVM løser problemet ved 1-norm blød margin.

Den primale repræsentation ved 1-norm blød margin

Optimeringsproblemet, der skal minimeres, er en kombination af længden af vægten $\overrightarrow{w}$ og slackvariablen ved 1-norm. Ved 1-norm er $k = 1$, så

$$\begin{aligned} \min \quad & \tfrac{1}{2}||\overrightarrow{w}||^2 + C\sum_{i=1}^{s}\xi_i^1 \\ \text{ubb.} \quad & y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \qquad \text{for } i = 1, ..., s. \end{aligned}$$

Strafværdien C vil som nævnt i afsnit 6.3.2.2 optimeres ved hjælp af tuning, som vi vil beskrive i afsnit 8.3. Dette gør sig også gældende ved 2-norm, som præsenteres senere.

Den primale Lagrangefunktion, udledt fra optimeringsproblemet ved 1-norm blød margin, er givet ved

$$L(\overrightarrow{w}, b, \overrightarrow{\xi}, \overrightarrow{\alpha}, \overrightarrow{\beta}) = \frac{1}{2}\langle \overrightarrow{w} \cdot \overrightarrow{w} \rangle + C\sum_{i=1}^{s} \xi_i - \sum_{i=1}^{s} \alpha_i [y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) - 1 + \xi_i] - \sum_{i=1}^{s} \beta_i \xi_i,$$

med $\alpha_i \geq 0$ og $\beta_i \geq 0$.

Den duale repræsentation for 1-norm blød margin

Den tilsvarende duale repræsentation findes ved først at differentiere, for bagefter at substituere ind i den primale Lagrangefunktion. Her differentierer vi med hensyn til de primale variable $\overrightarrow{w}$, $\overrightarrow{\xi}$ og b for at opfylde Karush-Kuhn-Tucker betingelserne,

$$\frac{\partial L(\overrightarrow{w}, b, \overrightarrow{\xi}, \overrightarrow{\alpha}, \overrightarrow{\beta})}{\partial \overrightarrow{w}} = \overrightarrow{w} - \sum_{i=1}^{s} y_i \alpha_i \overrightarrow{x}_i = \overrightarrow{0} \tag{44}$$

$$\frac{\partial L(\overrightarrow{w}, b, \overrightarrow{\xi}, \overrightarrow{\alpha}, \overrightarrow{\beta})}{\partial b} = \sum_{i=1}^{s} y_i \alpha_i = 0 \tag{45}$$

$$\frac{\partial L(\overrightarrow{w}, b, \overrightarrow{\xi}, \overrightarrow{\alpha}, \overrightarrow{\beta})}{\partial \overrightarrow{\xi}} = C - \alpha_i - \beta_i = 0 \tag{46}$$

og vi substituerer ind i den primale Lagrangefunktion, så

$$\begin{aligned} L(\overrightarrow{w}, b, \overrightarrow{\xi}, \overrightarrow{\alpha}, \overrightarrow{\beta}) &= \frac{1}{2}\left(\sum_{i=1}^{s} \alpha_i y_i \overrightarrow{x}_i\right)\left(\sum_{j=1}^{s} \alpha_j y_j \overrightarrow{x}_j\right) - \sum_{i=1}^{s} \alpha_i y_i \overrightarrow{x}_i \left(\sum_{j=1}^{s} \alpha_j y_j \overrightarrow{x}_j\right) \\ &\quad - \sum_{i=1}^{s} \alpha_i y_i b + \sum_{i=1}^{s} \alpha_i + \sum_{i=1}^{s} (C - \alpha_i - \beta_i)\xi_i \\ &= \sum_{i=1}^{s} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j \langle \overrightarrow{x}_i \cdot \overrightarrow{x}_j \rangle. \end{aligned} \tag{47}$$

Sidste udtryk er den duale objektfunktion, som vi kalder $W(\overrightarrow{\alpha})$, dvs.

$$W(\overrightarrow{\alpha}) = \sum_{i=1}^{s} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j \langle \overrightarrow{x}_i \cdot \overrightarrow{x}_j \rangle.$$

Af Karush-Kuhn-Tucker betingelserne får vi

$$\alpha_i[y_i(\langle \overrightarrow{x}_i \cdot \overrightarrow{w} \rangle + b) - 1 + \xi_i] = 0, \qquad i = 1, ..., s,$$
$$\beta_i \xi_i = 0, \qquad i = 1, ..., s.$$

Omskriver vi ligning (46),

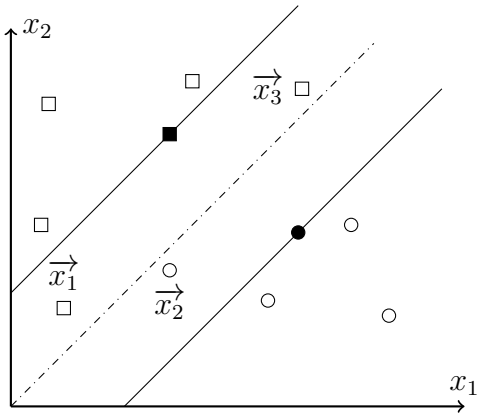
$$C - \alpha_i - \beta_i = 0 \Leftrightarrow \beta_i = C - \alpha_i,$$

kan vi skrive vores anden Karush-Kuhn-Tucker betingelse som,

$$\xi_i(C - \alpha_i) = 0, \qquad i = 1, ..., s,$$

og her vil $\xi_i > 0$ kun optræde når $\alpha_i = C$.

De observationer, hvis slack er forskellig fra nul, dvs. $\xi_i \neq 0$, kaldes for M-margin fejl. Dette fordi afstanden fra hyperplanet til observationer med slack er forskellig fra afstanden mellem hyperplanet, og de supportvektorer der ligger på marginen.



Figur 25

I figur 25 herover ved vi fra teorien om blød margin afsnit 6.3.2.2, at observationerne $\overrightarrow{x_1}$, $\overrightarrow{x_2}$ og $\overrightarrow{x_3}$ må have en slackværdi større end nul, da de ligger mellem marginen og det separerende hyperplan. Det ses også tydeligt, at afstanden fra observation $\overrightarrow{x_i}$ for $i = 1, 2, 3$ og ind til hyperplanet vil være kortere end afstanden for en hvilken som helst anden observation, der ligger på marginen og ind til hyperplanet.

Derfor kan vi ikke med sikkerhed vide at den geometriske afstand for en observation med slack større end nul, vil være lig $1/||\overrightarrow{w}||$, hvor observationer, for hvilke det gælder at $0 < \alpha_i < C$ og $\xi = 0$, har den geometriske afstand $1/||\overrightarrow{w}||$.

Bestemmelse af b ved 1-norm blød margin

Til forskel fra den hårde margin kan vi ikke finde to observationer, der ligger præcis på marginen ved at løse $\max_{y_i=-1}\langle\overrightarrow{w}^* \cdot \overrightarrow{x}_i\rangle$ og $\min_{y_i=1}\langle\overrightarrow{w}^* \cdot \overrightarrow{x}_i\rangle$, idet de observationer med slack vil have en henholdsvis højerere og lavere værdi af $\langle\overrightarrow{w}^* \cdot \overrightarrow{x}_i\rangle$ end de observationer, der ligger præcis på marginen.

Kigger vi derimod på den nye Karush-Kuhn-Tucker komplimentaritetsbetingelse, ved vi, at denne skal være aktiv når $\alpha_i \neq 0$, dvs. når $i \in sv$

$$\begin{aligned} y_i(\langle\overrightarrow{w} \cdot \overrightarrow{x_i}\rangle + b) - 1 + \xi_i^* &= 0 \\ \Rightarrow y_i(\langle\overrightarrow{w} \cdot \overrightarrow{x_i}\rangle + b) &= 1 - \xi_i^*. \end{aligned}$$

Vi har nu mulighed for at simplificere denne ved 1-norm, da vi ved at $\xi_i = 0$, når $0 < \alpha_i < C$. Dermed skal vi vælge b, således at følgende betingelse er overholdt.

$$\forall i \text{ hvor } 0 < \alpha_i < C : y_i(\langle\overrightarrow{w} \cdot \overrightarrow{x_i}\rangle + b) = 1.$$

Marginen ved 1-norm blød margin

Når vi tillader slack, betyder det, at ligning (37) ikke vil være opfyldt, hvormed vi ikke kan udtrykke marginen som ved den hårde margin. Udtrykket for det indre produkt $\langle\overrightarrow{w}^* \cdot \overrightarrow{w}^*\rangle$ kan dog simplificeres ved at benytte det, at alle $\alpha_i = 0,\ i \notin sv$, og at kernefunktionen kan indsættes i stedet for indre produkter, ligesom vi gjorde ved hård margin, da der ved blød margin ikke er nogen nye bibetingelser der afhænger af $\overrightarrow{x}$,

$$\langle\overrightarrow{w}^* \cdot \overrightarrow{w}^*\rangle = \sum_{i,j=1}^{s} y_i y_j \alpha_i^* \alpha_j^* K(\overrightarrow{x_i}, \overrightarrow{x_j}) = \sum_{i,j \in sv} y_i y_j \alpha_i^* \alpha_j^* K(\overrightarrow{x_i}, \overrightarrow{x_j}), \tag{48}$$

hvormed den geometriske margin kan udtrykkes ved

$$M = 1/||\overrightarrow{w}|| = \frac{1}{\sqrt{\langle\overrightarrow{w}^* \cdot \overrightarrow{w}^*\rangle}} = \left(\sum_{i,j \in sv} y_i y_j \alpha_i^* \alpha_j^* K(\overrightarrow{x_i}, \overrightarrow{x_j})\right)^{-1/2}.$$

7.2.3 2-norm blød margin

Vi vil nu præsentere, hvordan SVM vil løse optimeringsproblemet ved $k = 2$ i ligning (43), som vi kalder 2-norm blød margin.

Den primale repræsentation ved 2-norm blød margin

Da $k = 2 \Rightarrow \xi^2$, bliver optimeringsproblemet som følgende.

$$\begin{aligned} \min \quad & \tfrac{1}{2}||\overrightarrow{w}||^2 + C\sum_{i=1}^{s}\xi_i^2 \\ \text{ubb.} \quad & y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \end{aligned}$$

for $i = 1, ..., s$.

Her bemærker vi, at hvis $\xi_i < 0$ opfylder den første bibetingelse, så vil $\xi_i = 0$ også opfylde bibetingelsen, mens at $\xi_i = 0$ gør objektfunktionen strengt mindre end hvis $\xi_i < 0$. Alligevel betyder dette, at vi kan fjerne den positive begrænsning på ξ_i, der er en overflødig bibetingelse for problemet. Optimeringsproblemet for 2-norm blød margin fås dermed ved

$$\begin{aligned} \min \quad & \tfrac{1}{2}||\overrightarrow{w}||^2 + C\sum_{i=1}^{s}\xi_i^2 \\ \text{ubb.} \quad & y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) \geq 1 - \xi_i, \end{aligned}$$

for $i = 1, ..., s$.

Den primale Lagrangefunktion, udledt for optimeringsproblemet ved 2-norm blød margin, er givet ved

$$L(\overrightarrow{w}, b, \overrightarrow{\xi}, \overrightarrow{\alpha}, \overrightarrow{\beta}) = \frac{1}{2}\,||\overrightarrow{w}||^2 + \frac{C}{2}\sum_{i=1}^{s}\xi_i^2 - \sum_{i=1}^{s}\alpha_i\,[y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) - 1 + \xi_i],$$

hvor Lagrangemultiplikatoren $\alpha_i \geq 0$.

Den duale repræsentation for 2-norm blød margin

Den duale repræsentation findes ved først at differentiere Lagrangefunktionen fra den primale repræsentation med hensyn til $\overrightarrow{w}$, $\overrightarrow{\xi}$ og b

$$\frac{\partial L(\overrightarrow{w}, b, \overrightarrow{\xi}, \overrightarrow{\alpha}, \overrightarrow{\beta})}{\partial \overrightarrow{w}} = \overrightarrow{w} - \sum_{i=1}^{s} y_i \alpha_i \overrightarrow{x}_i = \overrightarrow{0}$$
$$\Rightarrow \overrightarrow{w} = \sum_{i=1}^{s} y_i \alpha_i \overrightarrow{x}_i \tag{49}$$

$$\frac{\partial L(\overrightarrow{w}, b, \overrightarrow{\xi}, \overrightarrow{\alpha}, \overrightarrow{\beta})}{\partial b} = \sum_{i=1}^{s} y_i \alpha_i = 0 \tag{50}$$

$$\frac{\partial L(\overrightarrow{w}, b, \overrightarrow{\xi}, \overrightarrow{\alpha}, \overrightarrow{\beta})}{\partial \overrightarrow{\xi}} = C\overrightarrow{\xi} - \overrightarrow{\alpha} = \overrightarrow{0}$$
$$\Rightarrow \overrightarrow{\xi} = \frac{\overrightarrow{\alpha}}{C}, \tag{51}$$

hvorefter de differentierede resultater substitueres ind i den primale Lagrangefunktion. Derved får vi følgende Lagrangefunktion,

$$\begin{aligned} L(\overrightarrow{w}, b, \overrightarrow{\xi}, \overrightarrow{\alpha}, \overrightarrow{\beta}) &= \frac{1}{2}\left(\sum_{i=1}^{s} \alpha_i y_i \overrightarrow{x}_i\right)\left(\sum_{j=1}^{s} \alpha_j y_j \overrightarrow{x}_j\right) + \frac{C}{2} \cdot \left(\frac{\overrightarrow{\alpha}}{C}\right)^2 \\ &\quad - \sum_{i=1}^{s} \alpha_i y_i \overrightarrow{x}_i \left(\sum_{j=1}^{s} \alpha_j y_j \overrightarrow{x}_j\right) - \sum_{i=1}^{s} \alpha_i y_i b + \sum_{i=1}^{s} \alpha_i - \frac{\overrightarrow{\alpha}^2}{C} \\ &= \sum_{i=1}^{s} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j \langle \overrightarrow{x}_i \cdot \overrightarrow{x}_j \rangle + \frac{1}{2C}\langle \overrightarrow{\alpha} \cdot \overrightarrow{\alpha} \rangle - \frac{1}{C}\langle \overrightarrow{\alpha} \cdot \overrightarrow{\alpha} \rangle \\ &= \sum_{i=1}^{s} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j \langle \overrightarrow{x}_i \cdot \overrightarrow{x}_j \rangle - \frac{1}{2C}\langle \overrightarrow{\alpha} \cdot \overrightarrow{\alpha} \rangle. \end{aligned}$$

Sidste udtryk er den duale objektfunktion, som vi kalder $\overrightarrow{\alpha}$, dvs.

$$W(\alpha) = \sum_{i=1}^{s} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j \langle \overrightarrow{x}_i \cdot \overrightarrow{x}_j \rangle - \frac{1}{2C}\langle \overrightarrow{\alpha} \cdot \overrightarrow{\alpha} \rangle,$$

hvor de tilhørende bibetingelser er,

$$\sum_{i=1}^{s} y_i \alpha_i = 0, \text{ og } \alpha_i \geq 0, \qquad \text{for } i = 1, ..., s$$

og den tilhørende Karusk-Kuhn-Tucker komplimentaritetsbetingelse er

$$\alpha_i[y_i(\overrightarrow{x}_i \cdot \overrightarrow{w} + b) - 1 + \xi_i] = 0, \qquad i = 1, ..., s.$$

Bestemmelse af b ved 2-norm blød margin

Vi kan, ligesom under 1-norm, bruge Karush-Kuhn-Tucker komplimentaritetsbetingelse og opstille de aktive bibetingelser for alle supportvektorer.

$$y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x_i} \rangle + b) = 1 - \xi_i^*, \qquad i = 1, ..., s,$$

hvor ξ_i^* kan udtrykkes ved (51). Vi får dermed, at vi skal vælge b, så følgende betingelse er opfyldt,

$$\forall i \in sv : y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x_i} \rangle + b) = 1 - \frac{\alpha_i^*}{C}. \tag{52}$$

Marginen ved 2-norm blød margin

Vi vil nu vise, at marginen kan skrives udelukkende udtrykt ved $\overrightarrow{\alpha}$ og C. Følger vi fremgangsmåden i udledningen fra den hårde margin fra ligning (38) til (41), og bruger ligning (32) og (33), samt det at $\forall i \notin sv : \ \alpha_i = 0$, hvor vi ligesom ved både hård margin og 1-norm blød margin indsætter kernefunktioner i stedet for indre produkter, så kan vi skrive $\langle \overrightarrow{w}^* \cdot \overrightarrow{w}^* \rangle$ som

$$\begin{aligned}
\langle \overrightarrow{w}^* \cdot \overrightarrow{w}^* \rangle &= \sum_{i,j=1}^{s} y_i y_j \alpha_i^* \alpha_j^* K(\overrightarrow{x_i}, \overrightarrow{x_j}) \\
&= \sum_{j \in sv} \alpha_j^* y_j \sum_{i \in sv} y_i \alpha_i^* K(\overrightarrow{x_i}, \overrightarrow{x_j}) = \sum_{j \in sv} \alpha_j^* (1 - \xi_j - y_j b^*) \\
&= \sum_{j \in sv} \alpha_j^* - \sum_{j \in sv} \alpha_j^* \xi_j - 0 \cdot b^* = \sum_{j \in sv} \alpha_j^* - \sum_{j \in sv} \alpha_j^* \xi_j
\end{aligned}$$

Ved at bruge ligning (51) kan vi nu skrive

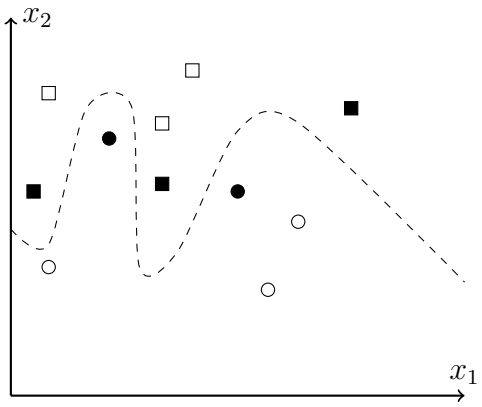
$$\begin{aligned}
\langle \overrightarrow{w}^* \cdot \overrightarrow{w}^* \rangle &= \sum_{j \in sv} \alpha_j^* - \sum_{j \in sv} \alpha_j^* \frac{1}{C} \sum_{j \in sv} \alpha_j^* \\
&= \sum_{j \in sv} \alpha_j^* - \frac{1}{C} \langle \overrightarrow{\alpha^*} \cdot \overrightarrow{\alpha^*} \rangle,
\end{aligned}$$

hvormed vi kan skrive den geometriske margin på følgende form

$$M = \left(\sum_{j \in sv} \alpha_j^* - \frac{1}{C} \langle \overrightarrow{\alpha^*} \cdot \overrightarrow{\alpha^*} \rangle \right)^{-1/2}.$$

7.3 Generaliseringsteori på SVM

I afsnit 6.5 redegjorde vi for den generelle generaliseringsteori i Machine Learning. I dette afsnit vil vi kigge på SVM's specifikke generaliseringsproblemer.



Figur 26

I figur 26 er den polynomiske kerne illustreret med supportvektorerne vist med sort udfyldning. Her er marginen ikke afstanden fra supportvektorerne til hyperplanet i det todimensionale rum men derimod afstanden mellem supportvektorerne og det lineære hyperplan i det transformerede rum.

Vi skal huske, at den separerende linje afbilder, hvordan det lineære hyperplan ser ud i det transformerede rum, og det betyder derfor ikke nødvendigvis, at der overfittes i figuren, selv om det ser sådan ud.

Vi skal dog stadig sikre os, at vi ikke har overfitting, altså at vi kan generalisere vores model fra et træningssæt til et testsæt uden at få helt andre klassifikationsrater. Den bedste måde ville være at teste på både vores trænings- og testsæt og sammenligne vores klassifikationsrater. Men et resultat udledt af V. Vapnik (*Vapnik, 2006*) siger, at ved *leave-one-out krydsvalidering*, som vi vil beskrive i afsnit 8.3.1, kan man kigge på antallet af supportvektorer, i form af følgende funktion

$$E[P(\text{fejl})] \leq \frac{E[\#\text{ support vektorer}]}{s}.$$

Denne funktion siger, at den øvre grænse for den forventede sandsynlighed af fejl er bestemt af det forventede antal af supportvektorer. Vi kan altså sikre os, at vores model generaliserer godt, hvis vi har et tilfredsstillende antal supportvektorer. Hvad der er et tilfredsstillende antal supportvektorer, findes der ikke nogen generel regel for, men der vil ofte i avancerede modeller være en højere andel supportvektorer end i de mere simple modeller. Her er det vigtigt at notere, at selv om et lavt antal supportvektorer sørger for god generalisering, kan det ikke vendes om. Man kan dermed ikke sige, at en model generaliserer dårligt udelukkende på baggrund af et højt antal supportvektorer. Et højt antal supportvektorer vidner mere om, hvor kompleks et problem vi har.

Robusthed overfor støj

En klar fordel ved at SVM er, at klassifikationsmodellen vil være robust overfor støj i datasættet. Når data klassificeres ved at maksimere den geometriske margin, fandt vi, at det kun er nogle af observationerne, der blev brugt til at bestemme hyperplanet og dermed klassifikationsmodellen. Ser vi på en outlier, der er klassificeret korrekt, og som ikke ligger på marginen, vil disse absolut ingen indflydelse have på hyperplanet, da observationens α vil antage værdien 0.

Med introduktionen af slackvariablen i blød margin betyder det også, at vi kan have outliere, som vil blive misklassificerede. Ved et fornuftigt valg af C vil de outliere, der ikke bør have indflydelse på klassifikationsmodellen blive klassificeret og dermed netop have en mindre indflydelse på det separerende hyperplan.

8 Algoritmen

I dette afsnit vil vi beskrive tankegangen bag en af de mest basale tilgange til, hvordan teorien bag SVM implementeres i en algoritme. Dette vil vi gøre ved forklarende tekst og ligninger samt med et rutediagram. Vi vil også komme ind på de fordele, som SVM har, i forhold til den beregningskraft implementeringen kræver. Vi vil ikke angive og kommentere på den specifikke kode, som vi vil anvende i programmet R.

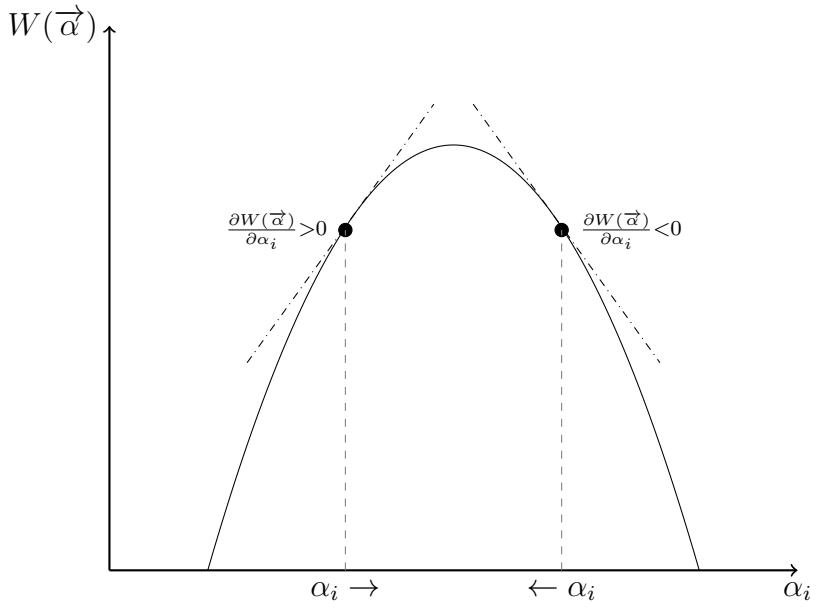
Algoritmen, som vi primært vil kigge på, hedder Gradient Ascent algoritmen, som er den mest simple algoritme, der samtidig illustrerer metoden bag mange andre tilgange til at implementere SVM. Vi vil dernæst komme ind på algoritmens svagheder og nævne forskellige forbedringsforslag som følge af disse.

8.1 Gradient Ascent algoritmen

Algoritmen initieres med et "gæt" på, at løsningen til det duale problem $\overrightarrow{\alpha}^0$ er lig nulvektoren. For hver iteration t forbedres løsningen ved at opdatere $\overrightarrow{\alpha}^t$. Hastigheden, hvormed $\overrightarrow{\alpha}$ opdateres, kaldes *læringsraten* η_i. Denne skal vælges, således at algoritmen konvergerer, hvilket vi vil beskrive senere i afsnittet. Ændringen for hver iteration er givet ved

$$\Delta\alpha_i^t = \eta_i \frac{\partial W(\overrightarrow{\alpha}^t)}{\partial \alpha_i}, \quad \eta_i \in \mathbb{R}^+, \ i = 1, ..., s. \tag{53}$$

Algoritmen øger værdien af objektfunktionen ved at opdaterer hver α_i en ad gangen. Da det duale optimeringsproblemet er konvekst, vil vi dermed aldrig få en suboptimal løsning.



Figur 27

Opdateringerne af α_i'erne gentages indtil kriteriet for, at den duale objektfunktion har nået sit maksimum, er opfyldt. Dette kriterie kaldes *stoppekriteriet* og kan defineres på forskellige måder. Vi vil opridse tre bud på disse i afsnit 8.1.2.

8.1.1 Rutediagram

Vi opdaterer α_i for hver iteration med dennes indflydelse på objektfunktionen $W(\overrightarrow{\alpha}^t)$'s værdi

$$\alpha_i = \alpha_i + \eta_i \frac{\partial W(\overrightarrow{\alpha}^t)}{\partial \alpha_i}, \tag{54}$$

hvor differentialet $\frac{\partial W(\overrightarrow{\alpha}^t)}{\partial \alpha_i}$ fås ved at differentiere vores duale objektfunktion fra ligning (42) hertil med hensyn til α_i, dvs.

$$\frac{\partial W(\overrightarrow{\alpha}^t)}{\partial \alpha_i} = 1 - y_i \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j}),$$

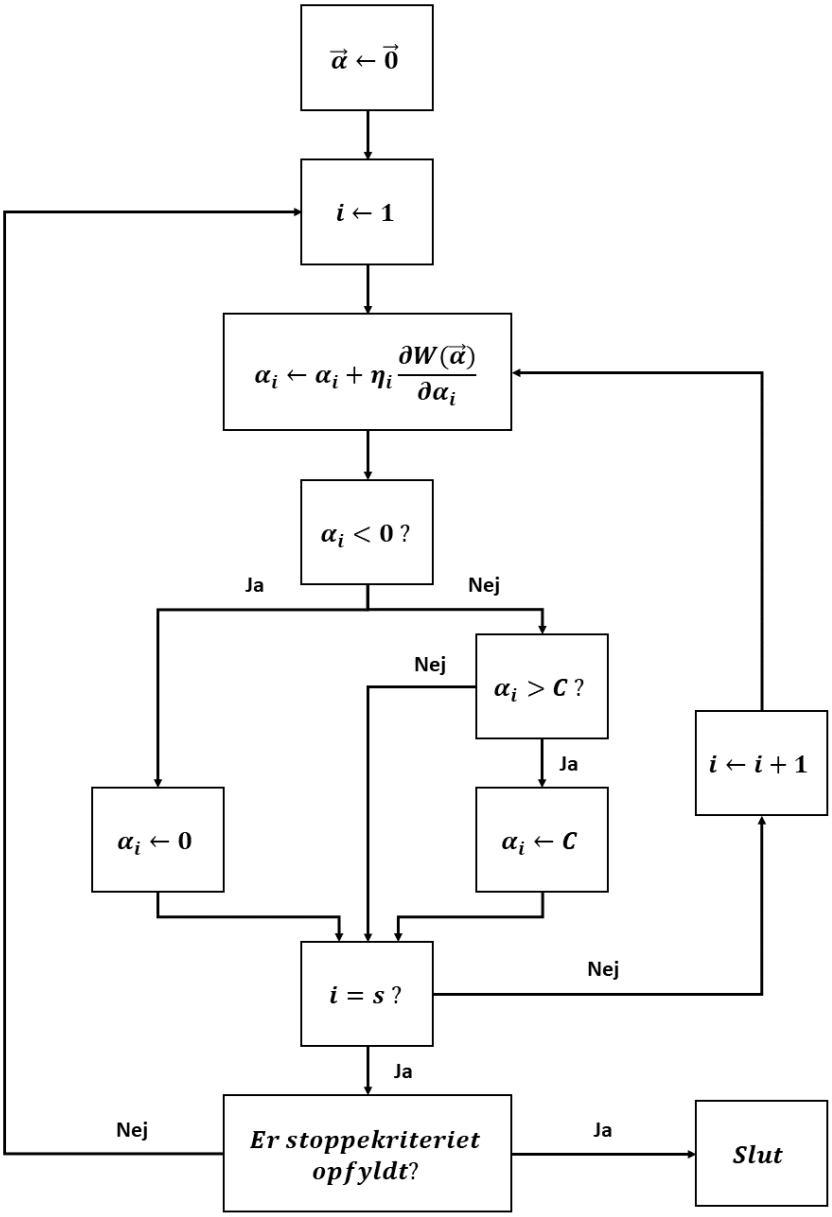
hvilket vi substituerer ind i ligning (54) og får

$$\Rightarrow \alpha_i = \alpha_i + \eta_i \left(1 - y_i \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j}) \right).$$

Idéen er altså, at vi opdaterer hvert α_i, indtil værdien af objektfunktionen $W(\overrightarrow{\alpha}^t)$ ikke kan forøges yderligere, hvilket er tilfældet, når stoppekriteriet er opfyldt.

Fra afsnit 6.3.2.2 ved vi, at $0 \leq \alpha \leq C$, og vi skal derfor sikre, at α_i ikke tager værdier under 0 ej heller større end C.

Rutediagrammet for algoritmen kan dermed illustreres som i figur 28.



Figur 28: Rutediagram for Gradient Ascent algoritmen

8.1.2 Stoppekriterier

Vi vil i det følgende redegøre for idéerne bag tre bud på stoppekriterier, dvs. hvor algoritmen skal stoppe og ikke længere søge efter en bedre løsning. Disse stoppekriterier er, når 1) objektfunktionen er maksimeret, 2) Karush-Kuhn-Tucker betingelserne er opfyldt, eller 3) når forskellen mellem den primale og den duale objektfunktion er nul.

Maksimer objektfunktionen

Optimeringsproblemet er løst, når objektfunktionen for det duale problem $W(\overrightarrow{\alpha})$ har nået sit maksimum. I programmeringen kræver dette, at vi definerer et *computernul*, $\epsilon \approx 0$. Når forøgelsen af værdien for vores objektfunktion er mindre end ϵ, afsluttes beregningerne. ϵ bestemmes ud fra en afvejning mellem præcision og beregningstid. Computernullet kan aldrig være præcis lig 0, da ændringen inden for programmering altid vil være *en anelse* højere end nul. Computernullet angiver altså, hvornår vi betegner en ændring for nul inden for programmering.

Karush-Kuhn-Tucker betingelserne

Karush-Kuhn-Tucker betingelserne sørger ligeledes for, at optimeringsproblemet er løst. Vi har her valgt kun at opstille betingelserne for 1-norm og 2-norm. Karush-Kuhn-Tucker-betingelserne har vi fra den duale repræsentation ved 1-norm blød margin (afsnit 7.2.2), som er

$$\begin{aligned} \alpha_i[y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) - 1 + \xi_i] = 0, \qquad & i = 1, ..., n, \\ \xi_i(C - \alpha_i) = 0, \qquad & i = 1, ..., n. \end{aligned}$$

Ligeledes har vi Karush-Kuhn-Tucker-betingelserne fra den duale repræsentation ved 2-norm blød margin (afsnit 7.2.3), som blot er

$$\alpha_i[y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) - 1 + \xi_i] = 0, \qquad i = 1, ..., n.$$

Betingelserne vil være opfyldt, når følgende er opfyldt med en præcision defineret ved vores computernul ϵ.

Ved 1-norm

$$y_i f(\overrightarrow{x_i}) = y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) \begin{cases} \geq 1, & \alpha_i = 0 \\ = 1, & 0 < \alpha_i < C \\ \leq 1, & \alpha_i = C. \end{cases}$$

Dette kommer af, at observationer, der opfylder $y_i f(\overrightarrow{x_i}) \geq 1$, er klassificeret korrekt og har ikke indflydelse på hyperplanet, derfor er $\alpha_i = 0$. Ved disse observationer bør $\xi_i = 0$ på baggrund af forudgående teori. Observationer, der opfylder $y_i f(\overrightarrow{x_i}) = 1$, er ligeledes klassificeret korrekt. Disse ligger på marginen og har indflydelse på hyperplanet, og derfor vil disse observationer medføre at $0 < \alpha_i < C$ jf. ligning (33). Slackvariablen for disse observationer må så have $\xi_i = 0$ for at opfylde Karush-Kuhn-Tucker betingelserne. Til sidst, for observationer der opfylder $y_i f(\overrightarrow{x_i}) \leq 1$, ved vi, at disse kan medføre slack, og at disse kan være misklassificeret, så $\xi_i \geq 0$. Det medfører, at $\alpha_i = C$ for at opfylde Karush-Kuhn-Tucker betingelserne.

Ved 2-norm gælder

$$y_i f(\overrightarrow{x_i}) = y_i(\langle \overrightarrow{w} \cdot \overrightarrow{x}_i \rangle + b) \begin{cases} \geq 1, & \alpha_i = 0 \\ = 1 - \alpha_i / C, & \alpha_i > 0, \end{cases}$$

som kommer af, at når observationerne opfylder $y_i f(\overrightarrow{x_i}) \geq 1$ må $\alpha_i = 0$ på samme måde som ved 1-norm. Når observationerne ligger på marginen eller indeholder slack, og dermed har en indflydelse på hyperplanet, så opfylder disse $y_i f(\overrightarrow{x_i}) = 1 - \alpha_i / C$, hvor $\alpha_i > 0$. Det vil desuden medføre, at slackvariablen implicit bliver $\xi_i = \alpha_i / C$. Stoppekriteriet er nu, at disse betingelser skal være opfyldt med en præcision defineret ved vores computernul ϵ.

Forskellen mellem den primale og duale objektfunktion

Som det tredje stoppekriterium ser vi på forskellen mellem den primale og den duale objektfunktion, da forskellen kun vil være nul i et optimum jf. afsnit 6.2. Vi begrænser os her til kun at opstille kriteriet for 1-norm.

Vi bruger igen vores computernul ϵ til at vurdere, hvornår denne forskel er lig nul, hvor optimeringsproblemet vil være løst, da vores optimeringsproblem er kvadratisk og konvekst.

Forskellen mellem den primale og den duale objektfunktion for 1-norm blød margin er givet ved

$$\begin{aligned}
&\frac{1}{2}\langle \overrightarrow{w} \cdot \overrightarrow{w} \rangle + C\sum_{i=1}^{s} \xi_i - W(\alpha) \\
&= \frac{1}{2}\langle \overrightarrow{w} \cdot \overrightarrow{w} \rangle + C\sum_{i=1}^{s} \xi_i - \sum_{i=1}^{s} \alpha_i + \frac{1}{2}\sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j K(\overrightarrow{x}_i \cdot \overrightarrow{x}_j). \qquad (55)
\end{aligned}$$

Hvis vi indsætter ligning (48) på pladsen for $\langle \overrightarrow{w} \cdot \overrightarrow{w} \rangle$ i ligning (55), får vi

$$\begin{aligned}
&\frac{1}{2}\sum_{i,j=1}^{s} y_i y_j \alpha_i^* \alpha_j^* K(\overrightarrow{x_i}, \overrightarrow{x_j}) + C\sum_{i=1}^{s} \xi_i - \sum_{i=1}^{s} \alpha_i + \frac{1}{2}\sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j K(\overrightarrow{x}_i, \overrightarrow{x}_j) \\
&\qquad = -\sum_{i=1}^{s} \alpha_i + \sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j K(x_i, x_j) + C\sum_{i=1}^{s} \xi_i \\
&\qquad = \sum_{i=1}^{s} \alpha_i - 2\sum_{i=1}^{s} \alpha_i + \sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j K(x_i, x_j) + C\sum_{i=1}^{s} \xi_i \\
&\qquad = \sum_{i=1}^{s} \alpha_i - 2\left(\sum_{i=1}^{s} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{s} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \right) + C\sum_{i=1}^{s} \xi_i. \qquad (56)
\end{aligned}$$

Når vi omskriver ligning (56), så får vi forskellen mellem den primale og den duale objektfunktion til

$$(56) \Rightarrow \sum_{i=1}^{s} \alpha_i - 2W(\overrightarrow{\alpha}) + C\sum_{i=1}^{s} \xi_i. \qquad (57)$$

Et rationelt stoppekriterie for algoritmen vil da være, når forskellen mellem den primale og den duale funktion er lig nul. Det betyder, at stoppekriteriet for algoritmen, med problemet hvor vi har 1-norm blød margin, bliver der, hvor resultatet i ligning (57) er lig vores computernul ϵ, dvs.

$$\sum_{i=1}^{s} \alpha_i - 2W(\overrightarrow{\alpha}) + C\sum_{i=1}^{s} \xi_i \leq \epsilon.$$

8.1.3 Konvergens

For at en algoritme kan implementeres er det nødvendigt, at den altid konvergerer. Ved Gradient Ascent algoritmen kan vi sikre dette ved at sørge for, at den afledte af objektfunktionen med hensyn til α_i vil gå mod nul. Vi kan skrive dette matematisk ved at kalde den iterativt opdaterede α_i for α_i^{it} defineret ved

$$\alpha_i^{it} = \alpha_i + \eta_i \left(1 - y_i \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j}) \right)$$

og kræve at

$$\frac{\partial W(\overrightarrow{\alpha}^{it})}{\partial \alpha_i} = 0. \tag{58}$$

Når ovenstående er opfyldt, vil vi befinde os i et globalt maksimum, da vores problem er et konvekst maksimeringsproblem. Sætter vi nu udtrykket for $\overrightarrow{\alpha}^{it}$ ind i udtrykket for $W(\overrightarrow{\alpha})$ differentieret med hensyn til α_i, får vi

$$\begin{aligned} \frac{\partial W(\overrightarrow{\alpha}^{it})}{\partial \alpha_i} &= 1 - y_i \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j}) \\ &\quad - \eta_i y_i y_i K(x_i x_i) \left(1 - y_i \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x}_i, \overrightarrow{x}_j) \right). \end{aligned}$$

Vi vil nu vise, at vi kan sætte

$$\eta_i = \frac{1}{K(\overrightarrow{x}_i, \overrightarrow{x}_i)},$$

således at kravet for konvergens fra ligning (58) er opfyldt. Indsætter vi ovenstående udtryk for η_i får vi kravet til at blive

$$\begin{aligned} \frac{\partial W(\overrightarrow{\alpha})}{\partial \alpha_i} &= 1 - y_i \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j}) \\ &\quad - y_i y_i + y_i y_i y_i \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j}). \end{aligned}$$

For $y_i = -1$ får vi

$$\frac{\partial W(\overrightarrow{\alpha})}{\partial \alpha_i} = 1 - (-1) \cdot \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j}) - (-1)^2 + (-1)^3 \cdot \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j})$$
$$= 1 + \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j}) - 1 - \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j}) = 0$$

og for $y_i = 1$ får vi

$$\frac{\partial W(\overrightarrow{\alpha})}{\partial \alpha_i} = 1 - \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j}) - 1 + \sum_{j=1}^{s} \alpha_j y_j K(\overrightarrow{x_i}, \overrightarrow{x_j}) = 0,$$

hvormed algoritmen altid konvergerer.

Konvergenshastighed

Det kan vises, at der eksisterer en fast rate τ, hvormed vi kommer tættere på den optimale værdi af vores objektfunktion $W(\overrightarrow{\alpha})$ for hver iteration. Denne rate kaldes *konvergensraten* og er i vores tilfælde givet ved

$$\exists \tau \in]0, 1[: \; W(\overrightarrow{\alpha}^*) - W(\overrightarrow{\alpha}^t + 1) \leq \tau\big(W(\overrightarrow{\alpha}^*) - W(\overrightarrow{\alpha}^t)\big).$$

Konvergensraten er dermed lineær. Dette er godt, fordi vi dermed er sikre på, at vi for hver iteration bevæger os i den rigtige retning. På figur 27 betyder dette, at vi, givet vi nærmer os optimum fra venstre, aldrig vil springe over på den højre side af optimum. Vores løsningsværdi nærmer sig altså hele tiden optimum med en positiv rate. At raten er fast, sikrer os, at vi i det mindste ikke vil nærme os optimum med en aftagende rate. Set i forhold til andre algoritmer er en lineær konvergensrate dog relativt langsom. Newton-Raphson algoritmen har eksempelvis ofte en kvadratisk konvergensrate.

Konvergenshastigheden kan forøges ved at ændre algoritmen til først at undersøge, hvilke observationer der påvirker objektfunktionen mest og opdatere disse først. En specifik løsning på dette vil vi nævne i afsnit 8.2. Vi vil dog først kigge på konvergensen af den polynomiske kerne samt Gradient Ascent algoritmens svagheder.

Den polynomiske kerne

Vi nævnte i afsnit 6.4.4.1 om den polynomiske kerne, at vi ved denne kerne risikerer, at algoritmen ikke vil konvergere.

Ved den polynomiske kerne, defineret ved ligning (24), risikerer vi, at algoritmen bruger meget beregningskraft, samt at algoritmen ikke konvergerer ved høje værdier af d, da

$$\lim_{d\to\infty} K(\overrightarrow{x_i}, \overrightarrow{x_j}) = (\gamma\langle\overrightarrow{x_i} \cdot \overrightarrow{x_j}\rangle + c)^d = 0, \qquad \gamma\langle\overrightarrow{x_i} \cdot \overrightarrow{x_j}\rangle + c < 1$$
$$\lim_{d\to\infty} K(\overrightarrow{x_i}, \overrightarrow{x_j}) = (\gamma\langle\overrightarrow{x_i} \cdot \overrightarrow{x_j}\rangle + c)^d = \infty, \qquad \gamma\langle\overrightarrow{x_i} \cdot \overrightarrow{x_j}\rangle + c > 1,$$

idet der er en øvre grænse for, hvor høje tal man kan arbejde med i programmering samt en nedre grænse defineret ved vores computernul ϵ.

8.1.4 Algoritmens svagheder

I Gradient Ascent algoritmen opdateres α_i'erne en ad gangen isoleret. Dette gør, at vi ikke kan overholde vores bibetingelse

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

samtidig med at ændre én α_i. Vi bliver nødt til at opdatere mindst to α_i'er ad gangen, for at kunne ændre på løsningen samtidig med at overholde bibetingelsen. Gradient Ascent algoritmen løser denne udfordring ved at fastsætte b'et, hvormed algoritmen ikke optimerer for b. Det er muligt at vise, at ved et fornuftigt valg af b på forhånd, vil dette kun have en minimal indflydelse på klassifikationensmodellens præstation.

En anden svaghed ved algoritmen er, at den kræver meget beregningskraft, idet alle α_i'erne skal opdateres for hver iteration.

Vi vil i næste afsnit angive kendte løsninger på disse udfordringer samt tilføjelser til algoritmen, der optimerer denne yderligere.

8.2 Tilføjelser til Gradient Ascent algoritmen

Vi vil først angive en tilføjelse, der kan nedsætte algoritmens beregningskraft betydeligt.

Det at beregne α_i'erne en ad gangen kræver en masse iterationer, som kræver meget beregningskraft. Nogle metoder afhjælper dette ved at gemme kernematricen, som er en kendt idé fra eksempelvis Newton-Raphson metoden. Dette kan dog blive et problem, når mængden af observationer bliver stor. Skal algoritmen implementeres på Big Data, vil datasættet indeholde millioner af observationer. Da kernen er en symetrisk nxn matrix, betyder det, at der nødvendigvis må gemmes $n(n+1)/2$ koefficienter. Med 8 byte allokeret til at gemme en variable af typen *double* som standard kræver en million observationer

$$\frac{8n(n+1)}{2}10^{-12} \approx 4 \text{ terabyte RAM},$$

da 1 byte svarer til 10^{-12} terabyte. At gemme kernematricen for Big Data er altså en umulighed for de fleste computere (*Steinwart & Christmann, 2008*).

Kigger vi i stedet på fortolkningen af α_i'erne, der skal opdateres, beskrevet i afsnit 6.2, fandt vi, at disse Lagrangemultiplikatorer kan tolkes som observation i's indflydelse på løsningen. Løsningen på udfordringen i forhold til beregningskraft er derfor at vælge kun at opdatere $\alpha_i^t > 0$. Disse er dermed supportvektorer i den t'te iteration og opdateres, indtil $W(\overrightarrow{\alpha})$ ikke kan forøges yderligere. I næste $t+1$'te iteration opdateres samtlige α_i'er, hvormed de nye supportvektorer i iteration $t+1$ findes.

For at overholde

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

skal vi som tidligere nævnt minimum opdatere to α_i'er ad gangen. I forhold til beregningskraft vil vi gerne opdatere færrest mulig observationer samtidig. Ved Sequential Minimal Optimisation, herefter SMO algoritmen, vælges derfor, for hver iteration α_i og α_j, og optimerer disse isoleret. Dette giver flere iterationer sammenlignet med

Gradient Ascent algoritmen, men det er muligt at vise, at SMO algoritmens iterationer kan beregnes i ét lukket udtryk, og derfor samlet set ofte vil kræve mindre beregningskraft. Vi kan også inden for SMO algoritmen spare beregningskraft ved at vælge α_i og α_j ud fra samme tankegang som ovenfor, nemlig at disse skal bidrage mest muligt til objektfunktionen.

En vigtig del af SVM er at finde de parametre, der giver det hyperplan, der bedst adskiller data. Hvordan dette gøres er ikke en del af algoritmen men kaldes *tuning*, og vil blive beskrevet i næste afsnit.

8.3 Tuning

Tuning handler om at finde de bedste parametre for en given kerne. Det er nyttigt at tune sine parametre, da dette kan forøge ens klassifikationsrate. Idéen er at lade algoritmen løbe igennem forskellige værdier for parametre i kernen på en fornuftig måde. Det negative ved tuning er, at der skal trænes mange modeller, og dette kan i mange tilfælde medføre en høj beregningstid. Der findes forskellige måder, hvorpå man kan tune sine parametre. Vi har dog valgt at begrænse os til en strategi kaldet gittersøgning (*Hsu, Chang & Lin*, 2016). Under tuningen er der brug for et mål for modellens præcision, således at man kan afgøre, hvilken værdi af parameteret man skal vælge. I gittersøgning bruges *krydsvalideringsraten* som dette mål.

8.3.1 Krydsvalidering

Krydsvalidering løser forskellige udfordringer såsom problemer med små datasæt eller generalisering. Disse anvendelser vil vi beskrive i dette afsnit

Vi vil på den ene side gerne træne en model med så meget data som muligt. På den anden side vil vi også gerne teste denne model på andet end træningsdata, fordi vi

ved, at en model med høj klassifikationsrate på træningssættet kan lide af overfitting, hvis den ikke også har en høj klassifikationsrate på testsættet. Krydsvalidering er en løsning på denne udfordring.

I krydsvalidering opdeles træningssættet i k lige store dele. Herefter træner man modellen på $k-1$ dele af træningssættet, mens man benytter den sidste del som testsæt. Dette gøres k gange, således at alle k dele af træningssættet har ageret testsæt én gang. Vi kalder dette testsæt for et *pseudo testsæt* for at skelne mellem dette og det egentlige testsæt. Resultatet er k klassificeringsrater på k pseudo testsæt. *Krydsvalideringsraten* er defineret som gennemsnittet af disse klassificeringsrater. Vælger vi en kerne og værdier for dennes parametre, er krydsvalideringsraten altså et mål for, hvor god modellen med de valgte parameterværdier er til at klassificere, idet der både er taget højde for ønsket om en høj klassifikationsrate på trænings- og testsættet.

Sættes $k = n$, hvor n er antallet af observationer, bygges modellen på alle observationer på nær én, som modellen testes på. Dette kaldes *leave-one-out* krydsvalidering. Fordelen ved dette er, at modellen trænes på mest muligt data, og samtidig testes på alle observationer - én ad gangen. I praksis er det dog et problem, at der skal trænes $k = n$ klassifikationsmodeller på træningssæt med $n - 1$ observationer i forhold til beregningskraft.

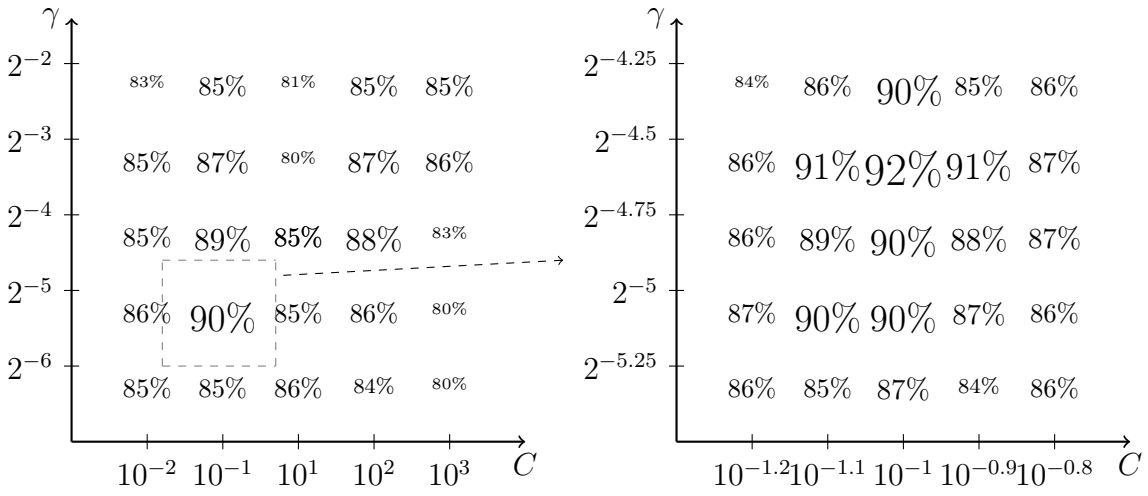
Sætter vi i stedet eksempelvis $k = 2$, betyder det, at vi træner modellen på halvdelen af data, hvorefter vi tester på den anden halvdel og derefter omvendt. Fordelen er da, at vi kun skal træne to modeller med et testsæt på $n/2$ observationer. Dette gør, at beregningskraften er lavere. Vi har dog dermed udvalgt en stikprøve på 50 % af data, som vi ikke kan være sikre på er repræsentativ udvalgt. Det mest populære valg af k i litteraturen er $k = 10$.

Krydsvalidering er især nyttigt, når datamængden er så begrænset, at det er uhensigtsmæssigt at sætte et testsæt til side i starten af analysen. Krydsvalidering kan dog med fordel bruges, selvom man har et testsæt som modellen testes på til sidst.

8.3.2 Gittersøgning

Gittersøgning er den mest simple metode til at finde de optimale parametre. Gittersøgningens simple tilgang til tuningen af parametre gør den dog også til den mest grundige og dermed også den mest krævende metode i forhold til beregningskraft. Algoritmen går nemlig ikke blot alle parameterværdier igennem men foretager også krydsvalidering for hver parameterværdi.

Ved gittersøgning for to parametre, eksempelvis C og γ, opstilles et gitter med værdier for det første parameter hen ad den ene akse, og værdier for det andet parameter hen ad den anden akse. Nu beregnes krydsvalideringsraten for hver kombination af parametre, således at de optimale værdier kan vælges. For at lette beregningskraften er det en fordel at foretage en gittersøgning med grove intervaller først og derefter indsnævre disse. Nedenfor er gittersøgningen illustreret for parametrene γ og C.



Figur 29

I ovenstående figur undersøges først krydsvalideringsraten for C og γ i intervallet henholdsvis $[10^{-2}; 10^{3}]$ og $[2^{-6}; 2^{-2}]$. Dernæst undersøges området omkring de parameterværdier med den højeste klassifikationsrate, som i eksemplet er $C = 10^{-1}$ og $\gamma = 2^{-5}$. Det er nu muligt at gøre intervallerne finere eller blot at stoppe gittersøgningen og dermed vælge $C = 10^{-1}$ og $\gamma = 2^{-4.5}$.

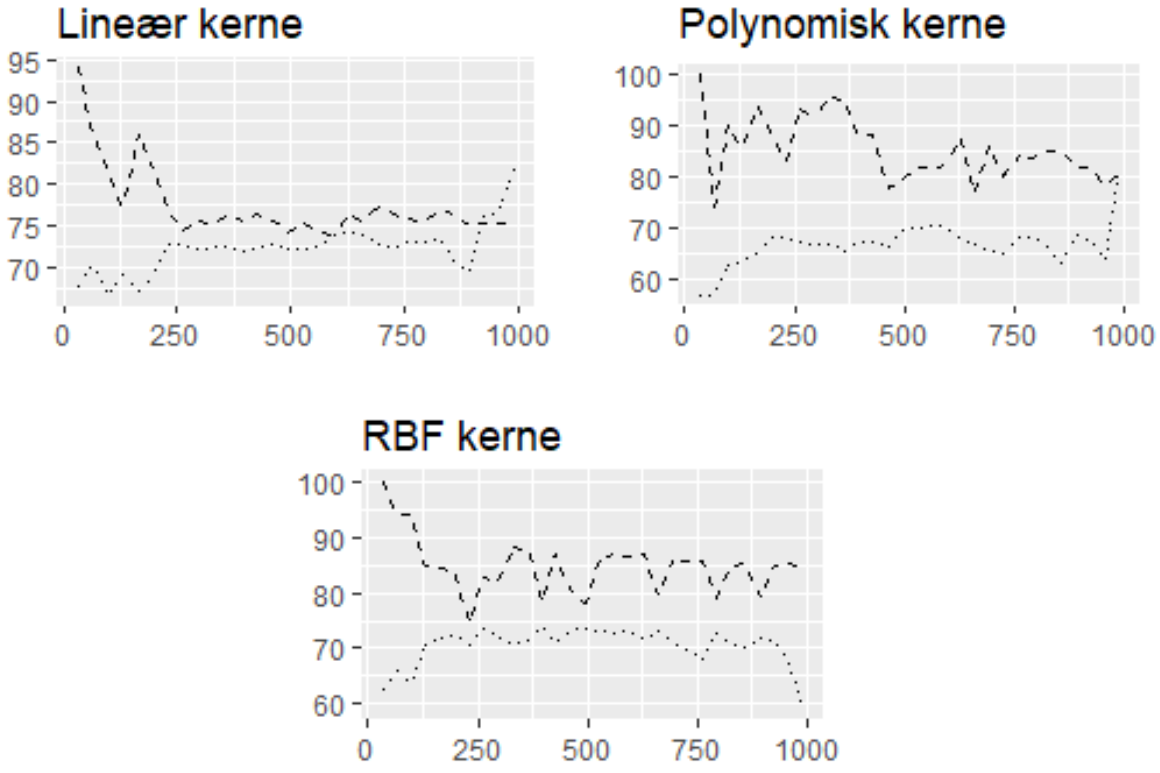
Gittersøgningen kompliceres en anelse i de tilfælde, hvor der er flere områder med relativt høje klassifikationsrater. I disse situationer er det nødvendigt at undersøge begge områder nærmere, før de optimale parameterværdier vælges.

Der findes som sagt andre metoder til at finde frem til de optimale parametre, som kræver mindre beregningskraft, men konsekvensen er en mindre grundig søgning. Da der ofte bruges kerner med to eller tre parametre, er det derfor normalt at vælge gittersøgning som metode for at finde de optimale parametre (Hsu, Chang & *Lin*, 2016).

9 Analyse

I dette afsnit vil vi udnytte teorien fra de forrige afsnit og anvende den på data, som er specificeret i afsnit 5. Koden, der er blevet benyttet til at finde resultaterne, er vedlagt digitalt i projektets bilag. Den software, der er benyttet, er programmet R.

Når data og de nødvendige pakker er indlæst, er det tid til at inddele data i et trænings- og testsæt. På denne måde kan vi undersøge, hvor godt vores model præsterer på et uafhængigt datasæt. Vi vil vælge størrelsen på trænings- og testsættet, ved at analysere *indlæringskurver*. Indlæringskurverne består af klassifikationsraten på henholdsvis trænings- og testsættet plottet mod antallet af observationer i træningssættet. Herudfra vurderes hvilken størrelse af trænings- og testsættet, der giver den bedste klassifikationsrate samt generalisering.



Figur 30

Kigger vi på figur 30, ser vi tre indlæringskurver ved den lineære, polynomiske og RBF-kernen. Den stiplede linje svarer til klassifikationsraten på træningssættet, mens den prikkede linje svarer til klassifikationsraten på testsættet.

Ud fra disse kan vi se, at der ikke er stor forskel på at vælge et træningssæt med mellem 500 observationer og et træningssæt med 800 observationer, da det primært er testsættets klassifikationsrate, vi ønsker at optimere. Vi vælger at sætte størrelsen på træningsssættet til 800 observationer, da det gængse i praksis er at sætte størrelsen til 80% af datasættet. Dette efterlader os med et testsæt på 216 observationer.

Vi opdeler vores data i et trænings- og testsæt tilfældigt, således at vi undgår bias. Vores træningssæt består af 397 aktive hoteller, og 403 hoteller der er gået konkurs. Vores testsæt består af 109 aktive hoteller og 103 hoteller, der er gået konkurs.

Fra figur 30 kan vi også se store forskelle på de forskellige kerner. I tilfældet med den lineære kerne ser vi, at når testsættet har en størrelse på over 250 observationer, ser modellen ud til at have en stabil klassifikationsrate og en god generalisering, da linjerne for henholdsvis trænings- og testsættet nærmer sig samme klassifikationsrate uden for stor variation. RBF-kernen og den polynomiske kerne ser dog hverken ud til at opnå stabil klassifikationsrate for både trænings- og testsæt eller at opnå god generalisering. Både den polynomiske kerne og RBF-kernen har tendens til overfitting. Dette kan skyldes, at parametrene ikke er tunet helt i læringskurven på figur 30. Vi vil derfor foretage en finere tuning, når vi analyserer de enkelte kerner.

Vi ønsker også at skalere vores data, da der er variable, hvor observationerne er i intervallet [0;25], mens andre variable er i intervaller op til flere millioner. Som tidligere nævnt er standardisering af data den mest anvendte metode, og det er også den normaliseringsmetode, vi anvender. Vi bruger en funktion, der tuner parametrene ved krydsvalidering og gittersøgning, og som også standardiserer data som standard. Det gør, at vi ikke selv er nødt til at standardisere, før vi kan begynde på analysen.

Vi vil nu træne klassifikationsmodellerne ved de forskellige kerner og efterfølgende sammenligne dem. Vi vil i alle tre kerner benytte en blød margin. Vi vil først kigge på den lineære kerne.

Lineær kerne

Hvis vi først laver en SVM model uden at tune vores strafparameter C men i stedet vælger det til at være $C = 10$, får vi en klassifikationsrate på 74.057% på vores testsæt. Vi vil nu se om denne kan øges ved at tune C.

Vi tuner strafværdien C ved at benytte en gittersøgning ved krydsvalidering med $k = 10$. Vi søger i intervallerne $C = 10^i$, hvor $i = (-3, -2, -1, 0, 1, 2, 3)$. Dette betyder, at vi tjekker C for henholdsvis, 0.001, 0.01, 0.1, 1, 10, 100 og 1000. Her får vi, at den bedste af disse værdier er $C = 1$. Vi ved nu fra teorien om gittersøgning, at den optimale strafværdi må ligge et sted mellem 0.1 og 10. Vi indsnævrer derfor søgningen et par gange, og får den optimale strafværdi til at være $C = 0.7$. Herefter bygger vi en model med den optimale strafværdi.

Modellen har 499 supportvektorer, hvilket vil sige, at over halvdelen af alle observationerne fra træningssættet har en indflydelse på hyperplanet. Som nævnt i afsnit 7.3 tyder det høje antal supportvektorer på, at vi har en kompleks model, men for at undersøge om det høje antal supportvektorer også kan skyldes overfitting, undersøger vi modellens klassifikationsrate på både trænings- og testsættet. Ved at teste modellen på træningssættet, som er det datasæt, vi har bygget modellen på, får vi en klassifikationsrate på 74.875%. Samtidig får vi, en klassifikationsrate på 75.472%, når vi tester på testsættet. Det er mere end 1% højere end vores analyse uden gittersøgning. At vores klassifikationsrater ligger så tæt på hinanden tyder på, at vores model ikke lider af overfitting og derfor er robust og god til at generalisere.

Vi vil nu undersøge, om vi kan få bedre resultater ved at bruge andre kerner.

Den polynomiske kerne

Den polynomiske kerne er den kerne, hvori flest variable skal tunes, hvilket også medfører, at denne analyse er den, der kræver mest beregningskraft, og den hvor gittersøgningen kan kræve flere skridt end i den lineære kerne. De parametre, der skal optimeres, er dem som vi nævnte i afsnit 6.4.4.1 nemlig γ, d og c samt strafværdien C, som er den samme som i den lineære kerne. Ligesom i den lineære kerne vil vi tune strafværdien ved gittersøgning og krydsvalidering med $k = 10$. Vi vil også tune γ med de samme værdier som C. Det optimale d vil vi tune ved at bygge tunede modeller for henholdsvis $d = 2,\ 3,\ 4$ og herefter undersøge, hvilket d der er bedst til at beskrive vores data. Modellen har ved flere gennemgange vist sig at være optimal i forhold til c, når $c = 0$, så den fastsætter vi til 0.

Ved at foretage gittersøgningen med krydsvalidering ved $k = 10$, hvor vi efterfølgende indsnævrer gitteret, kommer vi frem til følgende tre resultater for de tre forskellige værdier af d.

1. $\gamma = 5$, $C = 0.2$, $d = 2$. En model bygget med disse tunede parameterværdier giver en klassifikationsrate på 82.125% på træningssættet og en klassifikationsrate på 69.339% på testsættet samt et antal supportvektorer på 478.

2. $\gamma = 0.2$, $C = 40$, $d = 3$. Bygger vi vores model ud fra disse tunede parameterværdier, får vi en klassifikationsrate på 89.125%, når vi tester på træningssættet og en klassifikationsrate på 64.151% på testsættet. Herudover har vores model 457 supportvektorer.

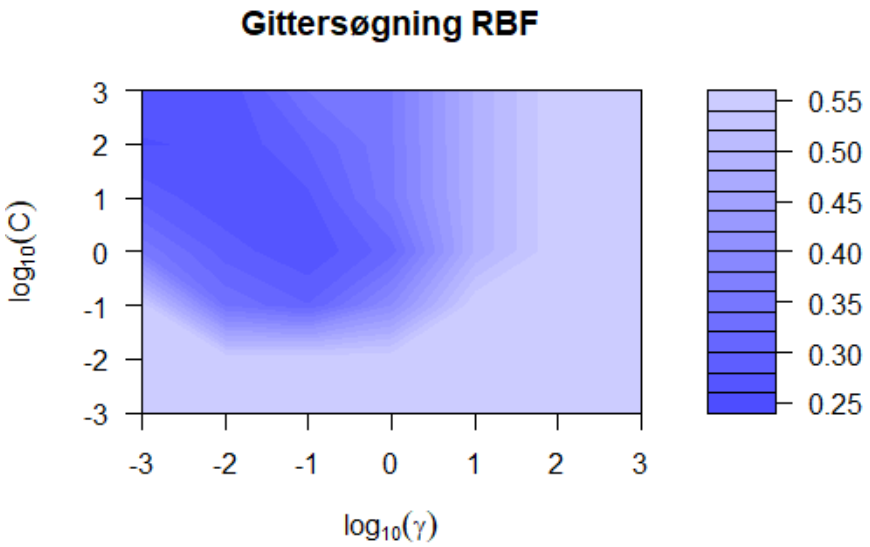
3. $\gamma = 0.6$, $C = 0.2$, $d = 4$. Disse tunede parameterværdier giver os en model med en klassifikationsrate på 86.125% på træningssættet og en klassifikationsrate på 68.369% på testsættet og indeholder 489 supportvektorer.

Disse resultater er meget forskellige fra de resultater, som vi opnåede med den lineære kerne, og vil blive diskuteret yderligere i afsnit 9.1.

RBF-kernen

Den gaussiske kerne er som tidligere nævnt den mest populære. I denne kerne skal vi ligesom i den polynomiske have tunet γ og et C. Modsat den polynomiske skal vi dog ikke tune et d.

Dette gøres som i de forrige analyser med en gittersøgning ved krydsvalidering med $k = 10$. Figur 31 viser den misklassifikationsrate, der fås ved forskellige værdier af γ og C, hvor misklassifikationsraten er defineret som én minus krydsvalideringsraten. Ud fra figuren, kan vi konkludere, at γ skal ligge omkring 0.001, og at C skal ligge omkring 100, da der er et lille område her, der er mørkere end resten. Herefter gør vi som i de forrige eksempler og kigger på et indsnævret interval omkring disse parameterværdier. Det medfører, at vi ender ud med et $C = 169$ og et $\gamma = 0.001$. Da vi bygger modellen på de fundne tunede parametre, får vi en klassifikationsrate på 76.625% for træningssættet, og en klassifikationsrate på 73.113% for testsættet.



Figur 31

9.1 Resultater

I dette afsnit vil vi se på resultaterne af vores analyse. Vi gennemgår først tabellen herunder, der opsummerer resultaterne af de forrige afsnit om analysen med de forskellige kerner.

Kerne	Lineær	Polynomisk $d = 2$	Polynomisk $d = 3$	Polynomisk $d = 4$	RBF
Træning	74.875%	82.125%	89.125%	86.125%	76.625%
Test	75.472%	69.339%	64.151%	68.396%	73.113%
#sv	499	478	457	489	505

Tabel 1

I tabel 1 ser vi de resultater, vi fandt frem til i analysen af vores data. Her kan vi se klassifikationsraterne for henholdsvis trænings- og testsæt, samt antallet af supportvektorer.

Det vigtigste vi kan konkludere ud fra tabellen, er resultaterne fra den polynomiske kerne, hvor vi for alle tre værdier af d observerer en stor forskel på klassifikationsraten mellem trænings- og testsættet. Dette tyder på at modellen lider af overfitting, som vi diskuterede i afsnit 6.5. Dette kan skyldes, at datasættet ikke bør adskilles af den polynomiske kerne, så når vi foretager en polynomisk transformation, hvad enten det er af anden, tredje eller fjerde grad, så fitter vi støj. Dermed generaliserer vores model dårligt fra træningssættet til tilsvarende data.

Ser vi på resultaterne fra den lineære kerne, kan vi se, at denne kerne har den højeste klassifikationsrate på testsættet, nemlig 75.472%, mens den ellers populære RBF-kerne har en lavere klassifikationsrate på 73.113%. Dette betyder, at data er bedst separeret ved en lineær kerne.

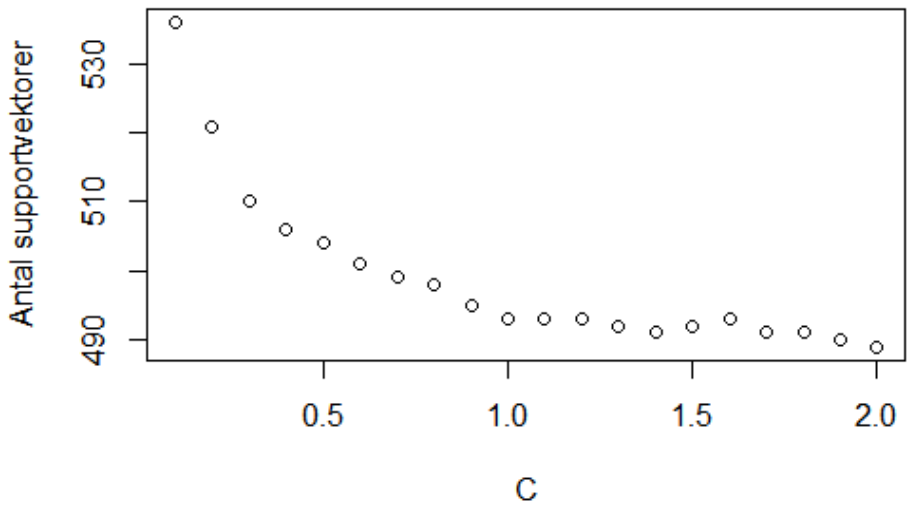
Fra tabel 1 ser vi yderligere, at andelen af supportvektorer i alle modellerne er over

50% af observationerne på træningssættet. Fra afsnit 6.5 ved vi, at hvis vi har en høj andel supportvektorer, kan vi risikere, at vores model generaliserer dårligt. I vores model ser vi dog, at generaliseringen er god ved den lineære kerne, da klassifikationsraten på trænings- og testsæt er næsten ens. Vi ser til gengæld en lille overfitting ved RBF-kernen, der har en forskel på omkring 3% imellem klassifikationsraten på trænings- og testsæt.

Resultaterne stemmer overens med vores forventninger baseret på figur 30, hvor vi forventede overfitting af den polynomiske og RBF-kernen, samtidig med at vi forventede en god generalisering af den lineære kerne.

Sammenhængen mellem C og supportvektorer

I afsnit 6.3.2.2 nævnte vi, at ved lave C-værdier, ville vi kunne forvente et høje antal supportvektorer. Vi træner nu en model med $C = 0.1, 0.2, ..., 2$.



Figur 32

Her ser vi som forventet, at antallet af supportvektorer er høje for små C-værdier. Resultatet er det samme for den polynomiske og RBF-kernen jf. figur 39 i bilag 13.2.

Beslutningsfunktionen

I vores model kan vi trække værdierne fra vores beslutningsfunktion ud, hvormed vi kan se hvilke værdier, der bør klassificeres i den ene og den anden klasse. Vi husker, at når $f(\overrightarrow{x}) \geq 0$, så tilhørte $\overrightarrow{x}$ én klasse, og når $f(\overrightarrow{x}) < 0$, så tilhørte $\overrightarrow{x}$ den anden klasse, samt at hvis $f(\overrightarrow{x}) = 1$ eller -1, så lå $\overrightarrow{x}$ på marginen. I vores analyse kan vi ud fra disse værdier afgøre, hvilke observationer, der er supportvektorer. Hvis vi sammenligner værdien af $f(\overrightarrow{x})$ med observationens klasse, kan nemlig vi se hvilke observationer, der er klassificeret korrekt og hvilke observationer der er misklassificeret.

Vi kan også trække sandsynligheder ud af vores model. Modellen er konstrueret således, at observationer der har over 50% sandsynlighed for at ende i en klasse, vil blive klassificeret i denne klasse. At have mulighed for at udtrække sandsynlighederne giver en fordel med hensyn til at benytte modellen i et virkelighedsscenarie. Hvis vi antager, at en investor er interesseret i at investere i et hotel, vil det være relevant ikke bare at vide, hvilken klasse hotellet er havnet i, men også hvor stor sandsynligheden er, for at denne prædiktion er korrekt. En investor vil nemlig alt andet lige hellere investere i en virksomhed, der eksempelvis har 90% sandsynlighed for ikke at gå konkurs, end i en, der har 60% for ikke at gå konkurs.

Observation	$f(\overrightarrow{x})$	P(Konkurs)
1	1.219	0.746
2	-1.8673	0.147
3	-0.336	0.412
4	0.883	0.683
⋮	⋮	⋮

Tabel 2

I tabel 2 kan vi se et udpluk af resultaterne for henholdsvis værdierne af beslutningsfunktionen og sandsynlighederne for konkurs. En højere værdi for $f(\overrightarrow{x})$ medfører som forventet en højere sandsynlighed for konkurs.

Ud fra informationen set i tabel 1 vil vi gå videre med den lineære kerne og udelukkende fokusere på disse resultater.

Ved den lineære kerne opnåede vi en klassifikationsrate på over 75% på testsættet. Dette betyder, at hvis vores model forudsiger, at en virksomhed vil gå konkurs, så vil den klassificere korrekt 75% af tiden.

Af tabel 3 kan vi se, hvordan vores model klassificerede vores observationer i testsættet. Vores model klassificerede 95 som aktive, hvoraf 76 af disse rent faktisk var aktive, mens 19 af dem var konkurs. Samtidig klassificerede vores model 117 som konkurs, hvoraf 84 var konkurs, og 33 var aktive. I alt var der altså 52 misklassifikationer og 160 korrekt klassificerede observationer.

Prædiktion	Aktiv	Konkurs	Total Prædiktion
Aktiv	76	19	95
Konkurs	33	84	117

Tabel 3

En klassifikationsrate på 75% er svær at fortolke, da den afhænger af forskellige ting som eksempelvis variablene eller kompleksiteten af den tilstedeværende data. Dette samt beregningstiden vil blive undersøgt i de næste afsnit.

9.2 Beregningskraft

For at en statistisk metode er brugbar, er det relevant at se på den beregningskraft, metoden kræver. Tager det eksempelvis flere dage at bygge modellen, vil metoden muligvis ikke være brugbar uanset dens præcision. Dette kriterie er især blevet relevant som følge af den generelle stigning i mængden af data, der anvendes til analyser. Det populære begreb Big Data handler netop blandt andet om udfordringen ved at analysere datasæt med millioner af observationer. Derfor er det ikke længere nok, at

man kan modellere en klassifikationsmodel med høj præcision. Desuden er levetiden for relevansen af resultater i højere og højere grad begrænset.

Dette skal tages i betragtning, hvormed den tid, det tager for den valgte metode at eksekvere, er relevant. Skal modellen anvendes på ny data eller endnu mere relevant, tilgangen bag metoden bruges til at udvikle en ny model, er det vigtigt, at implementeringen af metoden er effektiv. Vi vil derfor i dette afsnit undersøge den beregningskraft, det kræver at træne en klassifikationsmodel ved brug af SVM.

Vi vil nu undersøge, hvordan den beregningskraft SVM kræver påvirkes af antallet af transformationer, størrelsen på datasættet, antallet af variable, værdien for parametrene C, γ, c og d samt k i krydsvalideringen.

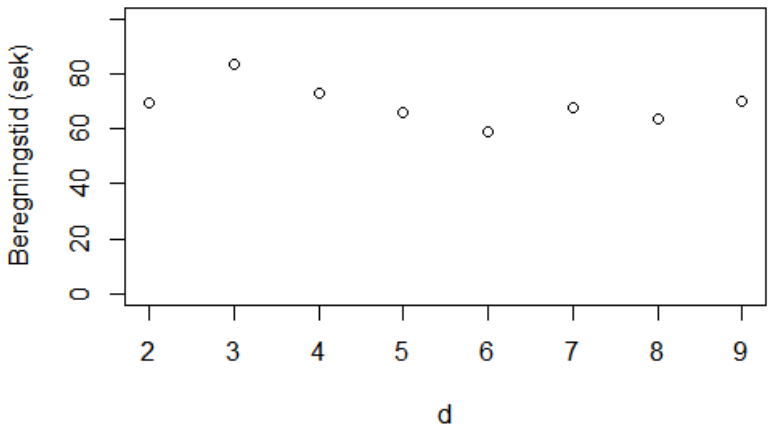
Kernetricket

Transformationer af de forklarende variable er ikke noget nyt inden for statistisk modellering. Det er derimod den måde hvorpå SVM transformerer, der giver metoden en fordel. SVM udnytter kernetricket til at foretage transformationerne indirekte, således at disse aldrig udregnes direkte som vist i eksempel 4. SVM har dermed mulighed for at foretage transformationer af de forklarende variable ind i et uendelig-dimensionalt rum, uden at det påvirker beregningskraften. Dette er som sagt en fordel, da sandsynligheden for at kunne adskille observationer stiger, når dimensionen af transformationen gør det.

Vi sammenligner beregningstiden for at udvikle en klassifikationsmodel med den lineære kerne, der ikke foretager nogen transformationer, og en model med RBF-kernen, der foretager transformationer ud i det uendelige rum jf. afsnit 6.4.4.3. Træner vi en model med den lineære kerne, med $C = 0.7$, tager det ca. 0.125 sekunder. Træningen af RBF-kernen med $C = 0.7$, og $\gamma = 0.1$ tager ca. 0.244 sekunder. Vi har her trænet modellerne 10 gange og taget gennemsnittet for at mindske indflydelsen af den støj, andre processer computeren kører ved siden af måtte have på resultatet. Vi ser altså, at det blot tager approksimativt dobbelt så lang tid, selvom vi foretager uendelig

mange observationer ved RBF-kernen.

Undersøger vi beregningstiden for graden d i den polynomiske kerne, får vi følgende graf for en gittersøgning ved $\gamma = 0.2, 0.4..., 1$ og $C = 0.1, 1, 10$.

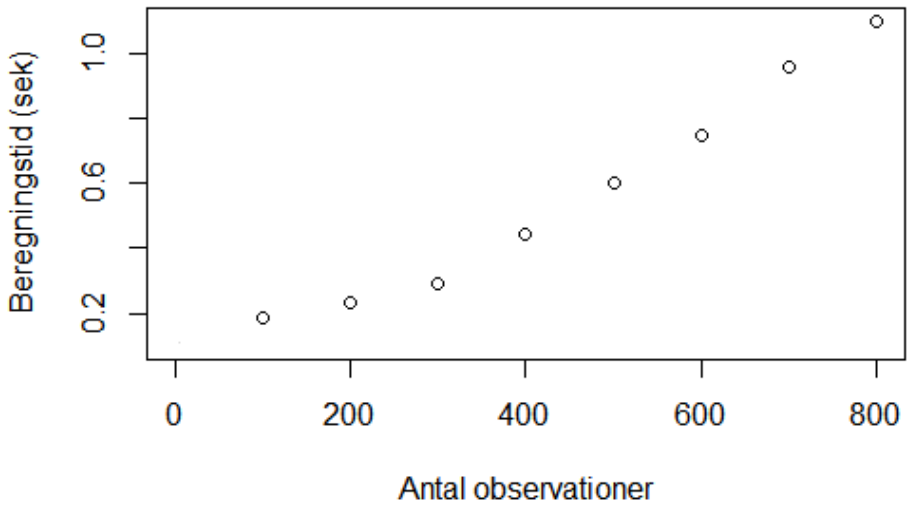


Figur 33

Her ser vi, at antallet af grader for polynomiet umiddelbart ikke påvirker beregningskraften. Vi ved fra afsnit 6.4.4.1, at dimensionen af det rum der projiceres ind i, stiger med d. Ovenstående figur viser altså igen, at kernetricket i SVM gør, at der ikke bruges beregningskraft proportionalt med dimensionen af det rum, der projiceres ind i.

Størrelsen på datasættet

Vi tester nu størrelsen af datasættets indvirkning på beregningskraften. Vi træner en model med den lineære kerne, hvor $C = 0.7$ på et datasæt med henholdsvis 100, 200, ..., 800 observationer, og får følgende beregningstider.

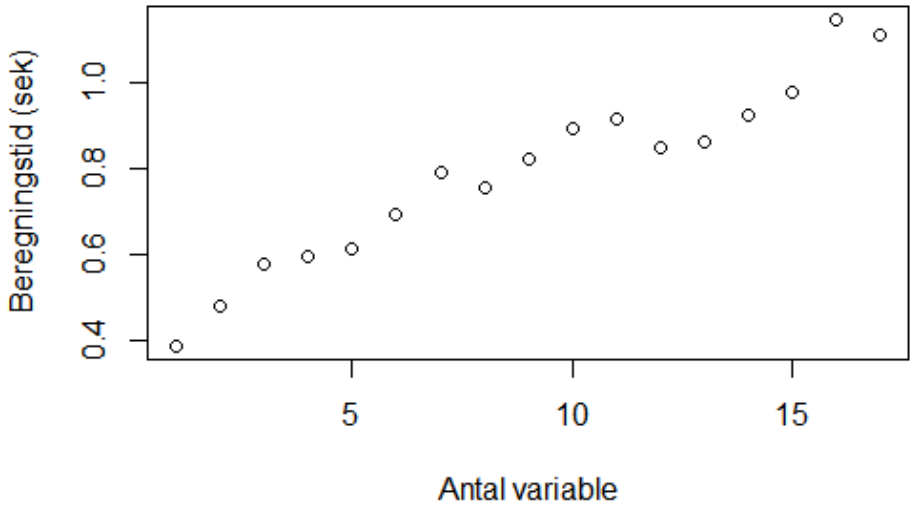


Figur 34

Vi ser dermed, at der er en tendens til et lineært forhold mellem beregningstiden og antallet af observationer i datasættet, der trænes på.

Antal variable

Vi vil nu undersøge, hvordan antallet af variable påvirker beregningstiden. Vi træner en model med den lineære kerne, med $C = 0.7$ og henholdsvis 1,...,17 observationer, og får nedenstående beregningstider.

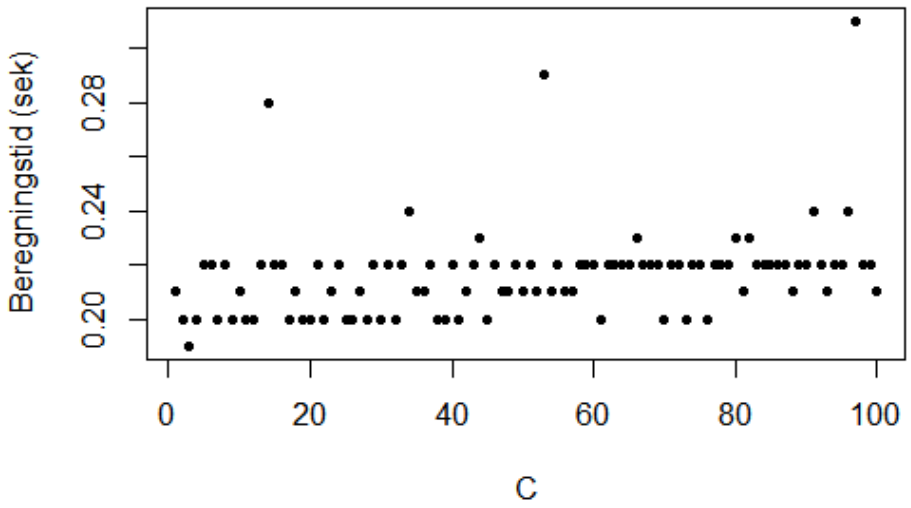


Figur 35

Igen ser vi en lineær tendens - her mellem beregningstiden og antallet af variable.

Parametrene C, γ, d og c

Det er svært at sige noget direkte om, hvordan værdierne af C,c, γ og d påvirker beregningskraften. Det afhænger af den specifikke fordeling data følger og af værdien af de andre parametre. Figur 33 viste, at man ikke kunne konkludere at beregningskraften er proportional med graden d af den polynomiske kerne. Kigger vi på C's indflydelse på beregningskraften, får vi følgende graf ved RBF-kernen, med $\gamma = 0.5$, og $C = 1, ..., 100$.

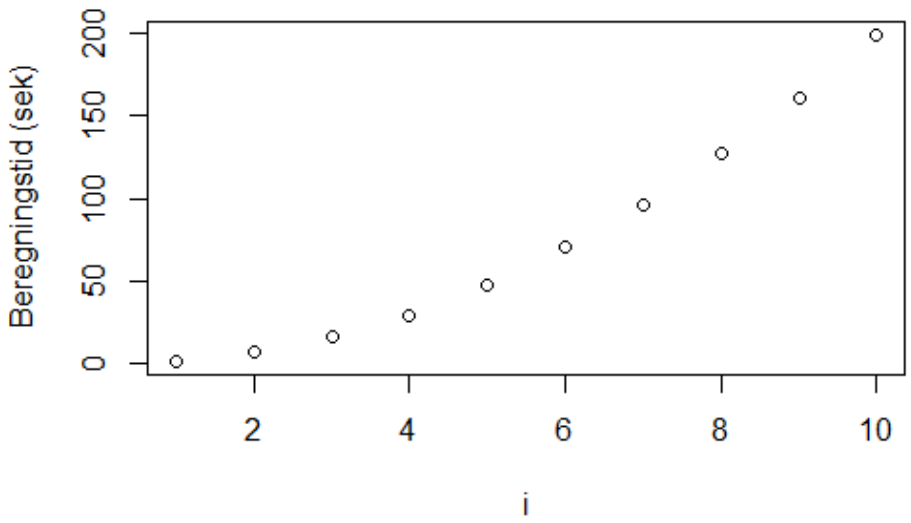


Figur 36

Figuren viser, at vi ikke kan konkludere et forhold mellem beregningstiden og parameteret C for RBF-kernen.

Gittersøgning

Vi undersøger nu den beregningstid, det tager at øge i ved en gittersøgning for RBF-kernen med $\gamma = 0.1 \cdot i$ og $C = 2 \cdot i$ for $i = 1, ..., 10$.

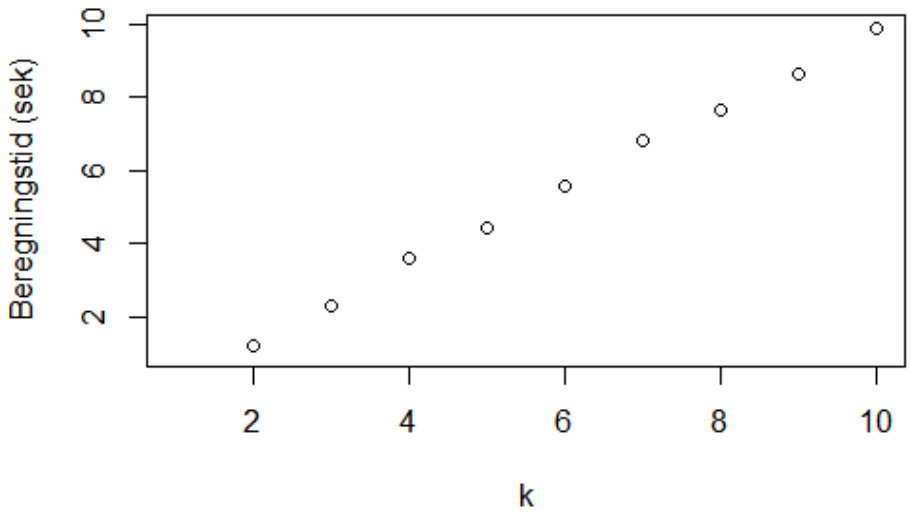


Figur 37

Hver gang i øges, øges gitteret med en ekstra række og søjle. Dette betyder at beregningskraften øges eksponentielt med antallet af parameterværdier, der søges i. En model med $\gamma = 0.1$ og $C = 2$ tog 1.95 sekunder at træne, mens en gittersøgning, med parameterværdier i intervallerne $\gamma = 0.1 \cdot 1, ..., 0.1 \cdot 10$ og $C = 2 \cdot 1, ..., 2 \cdot 10$, tog 198.59 sekunder.

Krydsvalidering

Vi ser nu på, hvordan valget af k i krydsvalidering påvirker den beregningskraft, gittersøgningen tager. Vi foretager her en gittersøgning for en model med den lineære kerne, med strafværdi på $C = \{0.1, ..., 1\}$, og med $k = \{2, ..., 10\}$, og får følgende beregningstider.



Figur 38

Som nævnt i afsnit 8.3.1, er der $s/2$ observationer i hvert træningssæt under en krydsvalidering med $k = 2$ og $s - s/10$ observationer i en krydsvalidering med $k = 10$. Af den åbenlyse lineære sammenhæng mellem beregningstiden og k i figuren ovenfor, lader det altså til, at den påvirkning, antallet af observationer modellen trænes på har på beregningskraften, er relativt lille i forhold til antallet af modeller, der skal trænes.

9.3 Ubalanceret data

I kapitel 5, påstod vi at ubalancerede datasæt vil have inflaterede klassifikationsrater. I dette afsnit vil vi undersøge, i hvilken grad dette er sandt.

Dernæst vil vi udvikle en klassifikationsmodel med den lineære og RBF-kernen, for at se hvordan denne præstere mod resultatet af vores oprindelige analyse.

Analyse af ubalanceret data

Vi starter med at lave et ubalanceret datasæt. Dette gør vi ved at tage alle 506 aktive observationer men blot 44 konkursramte observationer. Dermed får vi et datasæt med 8% konkursramte hoteller, og 92% aktive hoteller. Resultaterne af den nye analyse kan ses i tabel 4.

Kerne	Lineær	Polynomisk d=2	RBF
Træningssæt	92.44%	92.44%	92.44%
Testsæt	90%	90%	90%
#SV	79	96	68

Tabel 4

Som udgangspunkt ser de nye resultater lovende ud. Vi får generelt højere klassifikationsrater end vi gjorde i den oprindelige analyse. Denne nye analyse er dog lavet på et ubalanceret datasæt, hvilket betyder, at det er nemmere at få en højere klassifikationsrate. Hvis vi kigger på vores eksempel, hvor kun 8% af virksomhederne er konkursramte, kan vores model i teorien blot klassificere alle hoteller til at være aktive og dermed opnå en klassifikationsrate på 92%. I tabel 5 kan vi se, at dette netop er tilfældet for både trænings- og testsættet, hvor alle hoteller prædikteres til at være aktive.

Pred	Aktiv	Konkurs
Aktiv	416	34
Konkurs	0	0

Pred	Aktiv	Konkurs
Aktiv	90	10
Konkurs	0	0

Tabel 5

For at ændre denne klassifikation bliver vi nødt til at bygge en SVM model med høje værdier af C. Dette resulterede dog i lavere klassifikationsrater, og vi kan derfor konkludere, at træning på ubalanceret testsæt ikke er særlig effektivt i forhold til at bygge brugbare klassifikationsmodeller.

10 Diskussion & perspektivering

I vores analyse undersøgte vi teorien bag SVM ved at implementere denne i en analyse af konkurser på baggrund af finansielt data. En del af resultaterne fra vores analyse vil vi diskutere i dette afsnit.

Andre kerner

Vi undersøgte, hvordan SVM præsterede med forskellige kerner, og noterede, hvordan de forskellige kerner havde indflydelse på præstationen af klassifikationsmodellerne. Vi kunne her have gået videre med en undersøgelse af andre kerne - eksempelvis den populære Sigmoid kerne, og diskuteret disses præstation, i forhold til de kerner vi har undersøgt. Sigmoid kernen er en kerne, der ikke altid opfylder Mercers sætning, og netop derfor kunne det være interessant, at undersøge hvordan dette påvirker præstationen i forhold til at udvikle en god klassifikationsmodel.

Beregningskraft

Vi har i dette speciale redegjort for de vigtigste elementer i SVM, der henholdsvis forøger og reducerer beregningstiden. Her kunne det være interessant at sammenligne med andre metoder, for at kunne sige noget kvalificeret om SVM's præstation. Ved sammenligning med andre statistiske metoder er det dog nødvendigt at nærstudere teorien bag disse andre metoder først. Dette studie ville være relevant, idet vores anbefaling af SVM som metode afhænger drastisk af metodens præstation, set i lyset af de andre metoder der er til rådighed.

Vi viste, hvordan SVM både i teori og praksis ikke anvender beregningskraft proportionalt med antallet af dimensioner, for det rum som data projiceres ind i, på grund af kernetricket. Dette er interessant, fordi sandsynligheden, for at kunne separere data, er proportionalt med antallet af dimensioner, data projiceres ind i. Her kunne det være interessant at sammenligne med andre metoder, hvor transformationerne udregnes eksplicit. Det ville dermed blive muligt, at vurdere i hvilken grad denne implicitte transformation er en fordel sammenlignet med andre metoder. Ved ud-

førelsen af et sådant studie ville vi blive i stand til at sige noget om, hvor stort et datasæt og hvor mange variable man skal arbejde med, før denne fordel for alvor får betydning.

Vi har i vores analyse, figur 34 og 35, blot konstateret, at der er en tendens til et lineært forhold mellem beregningskraften og antallet af henholdsvis observationer og variable. Her kunne det igen være interessant at undersøge, om andre metoder ligeledes har et lineært forhold imellem disse, for at kunne afgøre, om disse forhold er en styrke for SVM set i forhold til andre metoder.

Med hensyn til parameterværdier og beregningskraft konkluderede vi, at man ikke kunne påvise et proportionalt forhold. Af figur 36 ses det, at specifikke værdier af parametrene kan give kraftige udslag for beregningstiden, hvilket afgjort er noget, man skal holde øje med, hvis man vælger at anvende SVM som metode.

Vi undersøgte også den beregningskraft, det krævede at tune parametrene ved gittersøgning. Figur 37 viste et eksponentielt forhold mellem beregningskraften og antallet af parameterværdier, der undersøges for. Øges antallet af variable og/eller observationer, vil dette især være noget, man skal være opmærksom på. Det er som nævnt i teorien muligt at løse denne udfordring delvist ved først at foretage en grov gittersøgning og dernæst indsnævre intervallerne for parameterværdierne. Det kunne dog være interessant at undersøge andre metoder for parametersøgning, således at vi ville blive i stand til at give et overblik over disses præcision og beregningstid set i forhold til gittersøgning.

Ubalanceret data

Vi ser fra analysen, at resultater med stærkt ubalanceret data er svære at fortolke. Dette vidner om, at vores beslutning om at smide aktive observationer væk, således at vores data blev balanceret, var en god beslutning. Vi kunne dog også have håndteret et ubalanceret datasæt på andre måder, eksempelvis ved at straffe slack fra den ene klasse hårdere end fra den anden.

Fortolkning af resultater

Anvender man SVM til at udvikle en klassifikationsmodel, bliver det svært at fortolke de enkelte variables indflydelse på prædiktionen. Dette vil man til gengæld kunne gøre ved at bruge traditionelle statistiske modeller. Denne viden ville kunne anvendes af hotelejere til at opnå indsigt i, hvilke finansielle forhold de skulle fokusere på at ændre og hvordan. Samtidig ville en sådan model med disse fortolkninger kunne give en indsigt i fænomenet konkurs, som ville være til gavn for resten af virksomhedens interessenter.

Et eksempel på en traditionel statistisk metode er logistisk regression. Logistisk regression benytter formlen

$$\frac{1}{1+e^{-Z}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_n x_n)}},$$

til at bestemme sandsynligheden for at en virksomhed går konkurs. En af fordelene ved logistisk regression er netop, at det er muligt at fortolke resultaterne. Ved logistisk regression kan man, ved hver enkelt virksomhed, finde et estimat for sandsynligheden for at denne går konkurs, og samtidig se hvilke variable der er signifikante i prædiktionen.

Principal Komponent Analyse

Ved et dybere fokus på implementeringen, kunne det være interessant at overveje, om principale komponenter, fundet ved en principal komponent analyse (PCA), vil være til gavn for SVM.

PCA er især en fordel, når data har mange variable, da det bliver muligt at beskrive disse enklere. Første principalkomponent beskriver mest varians i data, anden principal komponent beskriver næstmest varians osv., hvormed det bliver muligt at reducere antallet af variable.

Af figur 35 så vi, at beregningskraften steg med antallet af variable, hvormed en sådan reduktion af variable ville være nyttig ved store datasæt.

11 Konklusion

I dette speciale har vi redegjort for de teorier, der ligger til grund for SVM. Vi har beskrevet teorien fra optimeringsteori, hvor vi blandt andet beskrev den duale repræsentation af et optimeringsproblem, samt Karush-Kuhn-Tucker betingelserne for en optimal løsning til et kvadratisk optimeringsproblem.

Herefter fandt vi, at SVM forsøger at adskille data ved et lineært separerende hyperplan, der maksimerer den geometriske margin. Vi fandt herunder også ud af, at SVM har to metoder til at håndtere data, der ikke er lineært separabelt. Den ene metode er metoden ved blød margin, der tillader men straffer misklassifikationer ved at indføre en slackvariabel. Den anden metode introducerer brugen af kerner, der projicere data ind i et højere dimensionalt rum, hvor der er større sandsynlighed for at data er lineært separabelt. Vi fandt ud af, at det er optimalt at bruge en kombination af begge metoder, når data ikke er lineært separabelt.

Under teorien om kerner redegjorde vi for de krav Mercers sætning stiller for, at en funktion er en kerne. I forlængelse heraf præsenterede vi Hilbertrum, der tillader indre produkter i et uendelig-dimensionalt rum. Dette var nyttigt under brugen af den populære RBF-kerne. Ud over RBF-kernen undersøgte vi også den polynomiske og lineære kerne.

Efter introduktionen af kerner beskrev vi de problemer, der kan opstå i forbindelse med at operere i rum af en høj dimension. Disse problemer beskrev vi i generaliseringsteori, og vi kom frem til, at introduktionen af kerner kunne lede til overfitting. Vi forklarede begrebet overfitting, der består i, at en model ikke kan generaliseres til ny data.

Den sidste forudgående teori vi så på, var normalisering. Her kom vi frem til, at det var nyttigt at skalere data, da data som regel arbejder i forskellige størrelsesordner. Vi kom frem til at standardisering, der skalerer data, således at gennemsnittet bliver

nul og variansen bliver én, var den mest brugte form for normalisering.

Efter vi beskrev den forudgående teori, satte vi teorierne sammen for at definere og senere implementere SVM. Først opstillede vi SVM metoden med hård margin og senere med blød margin. Vi benyttede teorien fra optimeringsteori til at opstille en løsning for den maksimale geometrisk margin. Her fandt vi, at det var nyttigt at anvende den duale repræsentation af optimeringsproblemet, da dette kun afhang af én variabel, og at det muliggjorde anvendelsen af kerner.

I den bløde margin benyttede vi samme fremgangsmåde, dog med et ekstra led der skulle straffe misklassifikationer. Her skelnede vi mellem to forskellige metoder for blød margin, 1-norm og 2-norm, der straffer misklassifikationer forskelligt, alt efter hvor ekstreme observationer datasættet indeholder.

Til sidst i afsnittet om SVM fandt vi frem til nogle generaliseringsteoretiske resultater, der er unikke for SVM. Her kom vi frem til, at vi kunne bruge antallet af supportvektorer som en øvre grænse for fejl på testsættet ved brug af leave-one-out krydsvalidering.

Under implementeringen af SVM redegjorde vi for en algoritme ved navn Gradient Ascent algoritmen. Vi fandt, at denne algoritme iterativt opdaterer løsningen ved en fast læringsrate, indtil et stoppekriterie er nået. Vi gav desuden et bud på tre forskellige stoppekriterier. Herefter beskrev vi, hvordan vi valgte værdierne for parametrene i kernerne. Metoden vi anvendte var gittersøgning, der sammenligner klassifikationsmodellers nøjagtighed under forskellige parameterværdier, ved et mål kaldet krydsvalideringsraten.

Til sidst lavede vi en analyse, hvor vi illustrerede, hvordan vi praktisk kunne anvende SVM på et finansielt datasæt med to klasser - aktive og konkursramte hoteller. Her undersøgte vi resultaterne for de forskellige kerner.

Disse resultater viste god generalisering for den lineære kerne, mens vi så en tendens til overfitting ved den polynomiske kerne.

Vi analyserede også den beregningskraft SVM kræver, og hvordan forskellige elementer i metoden har indflydelse på denne. Her fandt vi frem til en tendens til et lineært forhold mellem beregningskraften og henholdsvis antallet af variable og observationer. Vi fandt også, at værdierne for de forskellige parametre har indflydelse på den beregningstid metoden kræver. Sidst i analysen fandt vi frem til, at det gav mening ikke at benytte strengt ubalanceret data.

12 Litteratur

Bernstein, M. N. *The Radial Basis Function Kernel*. PDF 2017.

Cawley, G. C. & Talbot, N. L. C. *On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation*. Artikel 2010.

Christianini, N. & Shawe-Taylor, J., *An Introduction to Support Vector Machines: and other kernel-based learning methods*. Cambridge University Press 2000.

Cortes, C. & Vapnik, V. *Support-Vector Networks*. Artikel 1995.

Database Orbis, Bureau van Dijk. Data trukket d. 17/04/2018.
Url: https://www.bvdinfo.com/en-gb/our-products/data/international/orbis

Eid1, H. F., Darwish, A., Hassanien, A. E. & Abraham, A. *Principle Components Analysis and Support Vector Machine based Intrusion Detection System*. Artikel 2010.

Hastie, T. & Tibshirani, R. & Friedman, J., *The Element of Statistical Learning*. 2. udgave. Springer Science + Business Madia, LLC 2009.

Hsu, C., Chang, C. & Lin, C. *A Practical Guide to Support Vector Classification*. Artikel 2016. Side 5-8.

James H Stock, J. H. & Watson, M. *Introduction to Econometrics*. International Edition, 3. udgave. Pearson Higher Education 2011.

Lay, D. C., *Linear Algebra and Its Application*. New International Edition 4. udgave. Pearson Education M.U.A 2013.

Pinch, R. *Cholesky factorization*. Encyclopedia of Mathematics, website 2016.
Url: https://www.encyclopediaofmath.org/index.php/Cholesky_factorization

Rudin, C. *Kernels*. MIT 15.097 Course Notes. Noter 2012.
Url:https://ocw.mit.edu/courses/sloan-school-of-management/
15-097-prediction-machine-learning-and-statistics-spring-2012/
lecture-notes/MIT15_097S12_lec13.pdf

Steinwart, I. & Christmann, A., *Support Vector Machines*. Springer Science + Business Madia, LLC 2008. Side 420-422.

Ng, R. *Support Vector Machines (SVMs)*. Website 2018.
Url: http://www.ritchieng.com/machine-learning-svms-support-vector-machines/

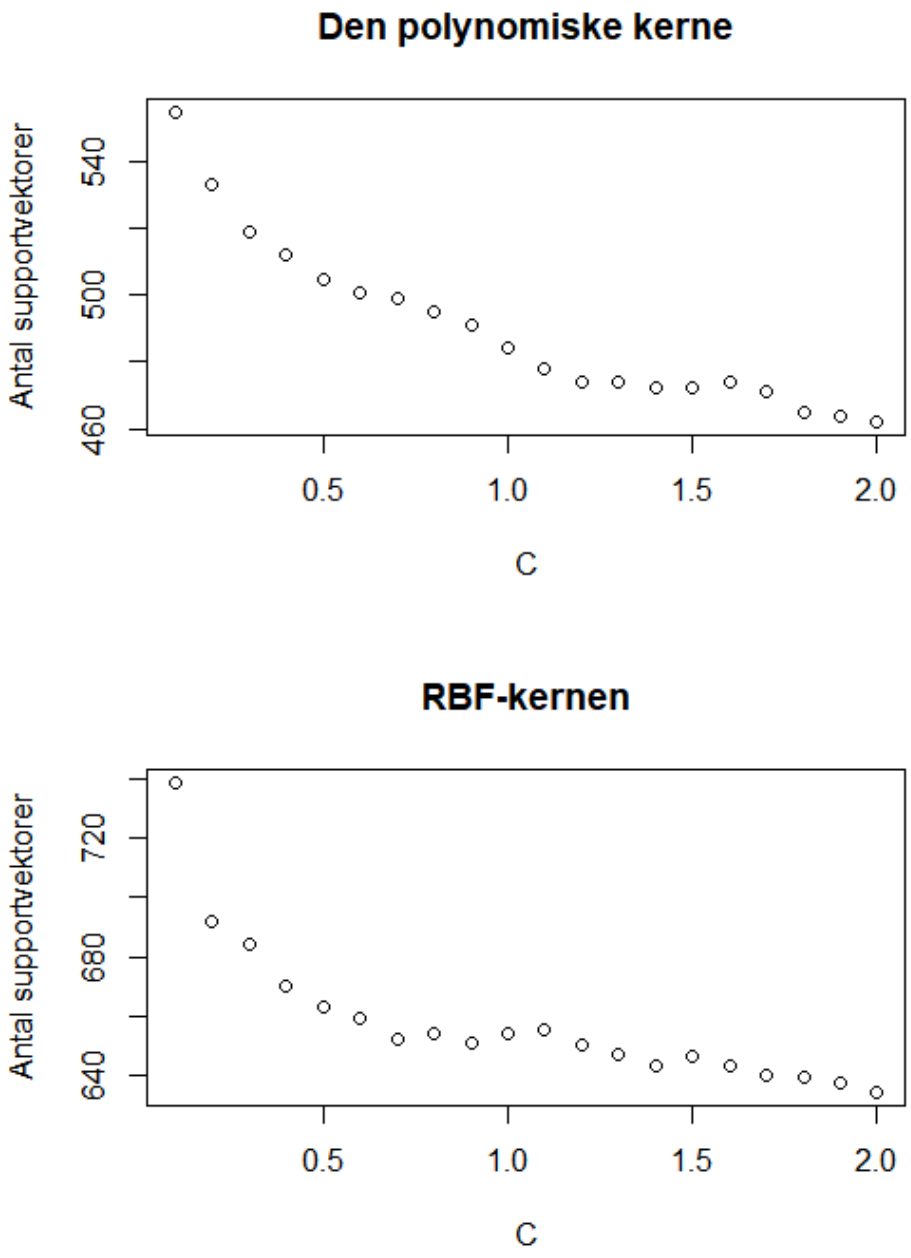
Vapnik, V. *Estimation of Dependences Based on Empirical Data*. 2. udgave, oversat udgave af 1982 udgaven. Springer Science + Business Media 2006.

13 Bilag

13.1 Bilag A - Valgte variable for analyse

1. Driftsindtægter (Operating Revenue)
2. Resultatopgørelse (P/L before tax)
3. Netto indkomst (P/L for period)
4. Pengestrøm (Cash flow)
5. Totale aktiver (Total assets)
6. Aktionærbevillinger (Shareholders funds)
7. Likviditets grad inkl varelager (Current ratio)
8. Overskudsgrad (Profit margin)
9. Soliditesgrad (Solvency ratio)
10. Totalkapitalforrentning ved resultatopgørelse (ROA using P/L before tax)
11. Egenkapitalforrentning ved Netto indkomst (ROE using Net income)
12. Totalkapitalforrentning ved Netto indkomst (ROA using net income)
13. Resultat før afskrivninger (EBITDA margin)
14. Resultat af primær drift (EBIT margin)
15. Lager omsætning (Stock turnover)
16. Kredit dage (Credit days)
17. Likviditets grad (Liquidity ratio)
18. Aktionærers Likviditets grad (Shareholders liquidity ratio)
19. Status (Status)

13.2 Bilag B - Plot



Figur 39: Antallet af supportvektorer plottet mod værdien af C. Modellerne er bygget med parameterværdierne $\gamma = 0.5,\ c = 0$