LUNDS UNIVERSITET

Att klassificera med Support Vector Machines

En introduktion från teori till analys

Alexandra Milton
Marcus Svensson
Kandidatuppsats i Statistik (15 hp)
Handledare: Björn Holmquist
VT 2018

Abstract

The purpose of this paper is to give a short introduction to support vector machines. The paper intends to cover the full process from theory to analysis in the binary classification case. The analysis intends to yield an understanding of how the method can be used in practice, and how models can be fitted in a way that maximizes the desired performance measurement.

This paper introduces the theory behind classification in the most basic case with perfect linear separability. Subsequently, methods to expand the model to manage more realistic scenarios of non-linear separability are introduced. Furthermore, the model selection process is detailed, and common performance measurements are proposed.

The dataset, Wisconsin Diagnostic Breast Cancer, is used for the analysis. Numerous models are trained to predict tumors as malignant or benign. The final model performs in line with predetermined requirements, correctly classifying as many malignant tumors as possible. The final model attains a sensitivity of 0.9836, which equals one false negative prediction.

Sammanfattning

Syftet med uppsatsen är att ge en kort introduktion till support vector machines. Uppsatsen ämnar täcka hela processen från teori till slutgiltig analys vid binär klassificering. Analysen ämnar ge en förståelse för hur metoden kan användas i praktiken och hur modeller kan anpassas efter bestämda önskemål med hjälp av olika prestandamått.

Uppsatsen introducerar teorin bakom hur perfekt linjärt separabla klasser behandlas samt metoder för att utveckla modellen till att hantera mer realistiska scenarion. Vidare presenteras tillvägagångssätt till modellanpassning samt prestandamått.

Datamaterialet, Wisconsin Diagnostic Breast Cancer, används för analys. Modeller anpassas i syfte att prediktera tumörer som elakartade eller godartade. Den slutgiltiga modellen presterar i linje med förutbestämda önskemål, att korrekt klassificera så många elakartade tumörer som möjligt. Den slutgiltiga modellen uppnår en sensitivitet på 0.9836 vilket motsvarar en elakartad tumör som felaktigt klassificerats som godartad.

Innehållsförteckning

1 Inledning

Maskininlärning är ett ständigt växande område och har under de senaste decennierna haft enorm genomslagskraft i samband med datorernas utveckling. Användandet av maskininlärning förekommer i alla möjliga situationer, från självkörande bilar och ansiktsigenkänning till situationer som konkurrerar med traditionella statistiska metoder som regression och klassificering.

Begreppet maskininlärning myntades under 1950-talet av Arthur Samuel. Genom applicering på spelet dam bidrog han till utvecklingen av att träna maskiner att lära sig från erfarenhet (McCarthy & Feigenbaum, 1990). Han menade att maskininlärning är en typ av artificiell intelligens i riktning att likna mänsklig kunskap (Samuel, 1959). Grundtanken bakom maskininlärning är att skapa självlärande system genom datoranvändning (Bell, 2015, s. 2). Dessa system tränas med hjälp av data för att genom bland annat mönsterigenkänning lära sig att utföra prediktioner.

Det förekommer vissa skillnader mellan maskininlärning och statistikområdet. Inom statistikområdet görs antaganden om en modell utifrån datamaterialet (Abu-Mostafa, Magdon-Ismail, & Lin, 2012a, s. 14). Modellen används för att skatta parametrar och utföra prediktioner. Inom maskininlärning används datamaterialet till att hitta en funktion som anpassas till observationerna för att prediktera den beroende variabeln (Breiman, 2001). Maskininlärning delas traditionellt upp i två områden: *supervised learning* och *unsupervised learning*. Supervised learning utgår från observationer med tillhörande output. Unsupervised learning utgår från observationer utan tillhörande output i syfte att få förståelse om strukturen i datamaterialet (Cristianini & Shawe-Taylor, 2000).

Klassificering är ett populärt användningsområde inom både maskininlärning och statistikområdet. Att klassificera är en naturlig utmaning som människor angriper dagligen både medvetet och omedvetet. Begreppet ”att klassificera” innebär att dela in individer i två eller flera grupper utifrån specifika indelningsgrunder (Nationalencyklopedin, 2017). Exempel på klassificeringssituationer är när kreditföretag vill skilja på låg- eller högriskkunder, när vädret bedöms som varmt eller kallt, eller när läkare vill bedöma tumörer som god- eller elakartade. Logistisk regression och diskriminantanalys är exempel på metoder inom statistikområdet för att lösa klassificeringsproblem. Den här uppsatsen angriper klassificeringsproblematiken med maskininlärning, med hjälp av en metod som benämns Support Vector Machines (SVM).

Grunden till SVM lades av Vladimir Vapnik år 1965. Han utvecklade en metod att linjärt separera klasser med största möjliga marginal. Under 1990-talet utvecklades metoden till att även kunna hantera icke-linjärt separabla klasser (Cortes & Vapnik, 1995).

Syftet med uppsatsen är att förmedla en introduktion till SVM och applicera metoden på ett utvalt datamaterial med avgränsning till binär klassificering. Uppsatsen ämnar täcka hela processen från teori till slutgiltig analys. Det datamaterial som används för att illustrera metodiken, Wisconsin Diagnostic Breast Cancer, är insamlat genom biopsier vid Wisconsins universitetssjukhus mellan åren 1989 och 1991. Dessa tester har sedan bildanalyserats och givit grunden för de mätvärden som kommer analyseras. Analysen utförs i R med hjälp av paketet *e1071*.

2 Introduktion till SVM

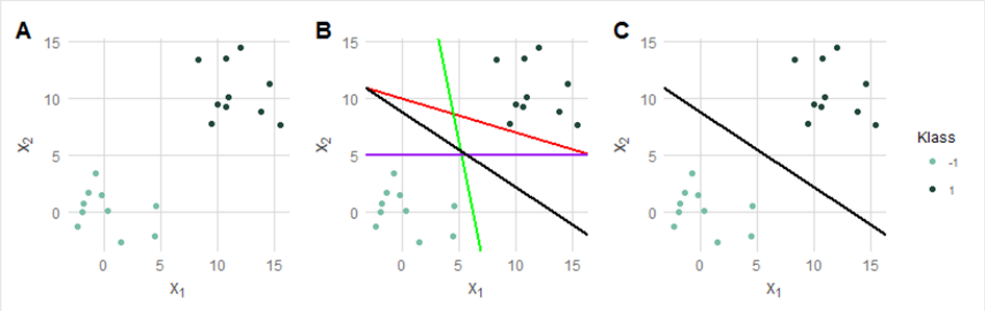
I detta kapitel presenteras en introduktion till SVM och hur metoden fungerar i generella drag. Därefter presenteras några grundläggande begrepp, samt styrkor och svagheter med metoden.

2.1 Vad är SVM?

SVM är en klassificeringsmetod som med hjälp av träningsdata skapar beslutsunderlag för klassificering av nya observationer till förutbestämda klasser (Abu-Mostafa, Magdon-Ismail, & Lin, 2012b, ss. 1-3).

Punkterna i Figur 2.1A representerar observationer från ett simulerat datamaterial. Punkternas olika färg illustrerar observationernas klasstillhörighet. De två klasserna benämns -1 respektive $+1$ vilket förklaras närmare i Kapitel 3. Datamaterialets två klasser kan separeras linjärt på ett oändligt antal sätt. Figur 2.1B illustrerar några exempel på hur olika beslutsgränser kan dras för att uppnå detta. Syftet med SVM är att hitta den beslutsgräns som separerar klasserna med största möjliga marginal (Hastie, Rosset, Tibshirani, & Zhu, 2004, s. 2). Detta uppnås genom att identifiera de punkter som bestämmer det kortaste avståndet mellan klasserna. Punkterna används som hjälpmedel för att separera materialet. Beslutsgränsen placeras mellan dessa punkter så att det rätvinkliga avståndet från punkterna till beslutsgränsen blir lika för alla punkter. Detta genererar största möjliga avstånd mellan klasserna och beslutsgränsen (Gunn, 1998). Denna beslutsgräns benämns det optimala hyperplanet. Det optimala hyperplanet för det simulerade datamaterialet illustreras i Figur 2.1C.

I exempelmaterialet i Figur 2.1 har variablerna samma skala, ingen av variablerna har en variationsbredd som klart dominerar över andra variabler. Klasserna är



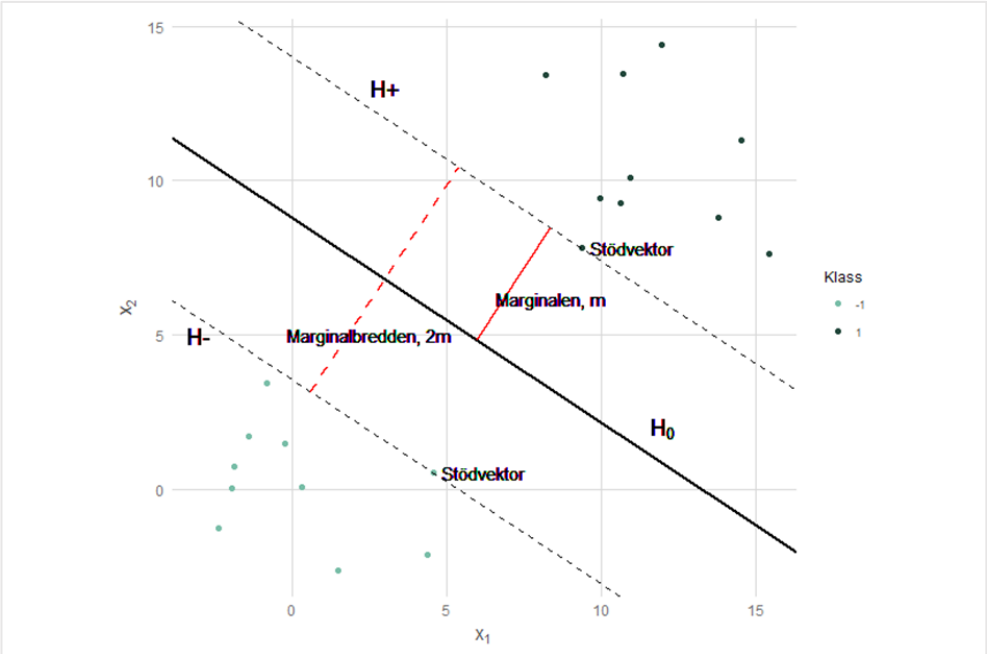
Figur 2.1 *Exempel på hyperplan som separerar ett datamaterial. I Figur B illustreras flera hyperplan som separerar klasserna perfekt. I Figur C illustreras det hyperplan som separerar klasserna med största möjliga marginal. Detta hyperplan benämns det optimala hyperplanet.*

dessutom perfekt linjärt separabla. I praktiken är detta sällan fallet. Variabler med varierande skalor kan leda till att de med stor variationsbredd dominerar de med liten variationsbredd. Detta kan även öka beräkningsbelastningen vid modellanpassning. I praktiken standardiseras nästan alltid variablerna före modellanpassning (Hsu, Chang, & Lin, 2003)

Två metoder kan användas för att anpassa ett optimalt hyperplan när klasserna inte är linjärt separabla. Dessa innebär applicering av en *mjuk marginal* och/eller en *kärnfunktion*. En mjuk marginal innebär förenklat att observationer tillåts ligga fel vid anpassning av det optimala hyperplanet (Abu-Mostafa et al., 2012b, ss. 40-41). Att applicera en kärnfunktion innebär att observationerna betraktas i en ”införd” högre dimension (Bishop, 2006, ss. 291-292). Detta skapar förutsättningar att kunna separera datamaterialet linjärt i det nya rummet av högre dimension. Dessa metoder förklaras mer detaljerat i Kapitel 4.

2.2 Grundläggande begrepp

De observationer som bestämmer förutsättningarna för var beslutsgränsen ska placeras benämns support vectors, därav namnet SVM. I följande kapitel används den svenska översättningen stödvektorer.



Figur 2.2 *En SVM anpassad till ett linjärt separabelt datamaterial. Här visualiseras marginalen, m, det optimala hyperplanet H_0 samt de parallella hyperplanen H_+ och H_- som skär stödvektorerna.*

Den beslutsgräns som separerar datamaterialets två klasser med största möjliga marginal benämns det optimala hyperplanet. Det optimala hyperplanet betecknas med H_0. Begreppet hyperplan används eftersom datamaterial vanligtvis består av fler dimensioner än två. I de fall då enbart två dimensioner förekommer, är den korrekta termen för beslutsgränsen en rät linje. Vid enbart en dimension motsvaras detta av en punkt (Abu-Mostafa et al., 2012b, ss. 2-3,7-8).

Marginalen, m, definieras som det rätvinkliga avståndet mellan det optimala hyperplanet och dess närmsta observation. Två parallella hyperplan skapas på vardera sida om det optimala hyperplanet och går genom de observationer som ligger närmast, stödvektorerna. Dessa parallella hyperplan betecknas med H_+ respektive H_-. Eftersom det optimala hyperplanet bestäms med förutsättningen att datamaterialet separeras med lika stor marginal till båda klasserna, definieras marginalbredden som $2m$ (Burges, 1998, s. 8). Ovanstående begrepp illustreras i Figur 2.2.

2.3 Styrkor och svagheter

2.3.1 Styrkor

Många datamaterial kräver att problemlösning sker i flera dimensioner. Styrkan hos SVM är att metoden dels kan appliceras på material bestående av ett stort antal dimensioner men också anpassas efter både linjära som icke-linjära klassificeringsproblem genom att applicera en mjuk marginal eller kärnfunktion på materialet (Abu-Mostafa et al., 2012b, s. 19).

Till skillnad från många statistiska metoder behöver inte SVM anpassas efter fördelningsantaganden på grund av att inga statistiska antaganden görs gällande den underliggande fördelningen i materialet (Kecman, 2001, s. 24).

Modellen är inte speciellt känslig för outliers. Genom att välja rätt parametrar kan effekten av outliers dämpas. Generaliseringsförmågan hos en modell, det vill säga hur väl modellen presterar vid prediktion av framtida observationer, påverkas av de val som görs gällande modellens parametrar. Om parametrarna anpassas väl, är SVM en robust metod med hög generaliseringsförmåga (Steinwart & Christmann, 2008, s. 12; Abe, 2010, s.59).

2.3.2 Svagheter

Att välja modellparametrar kan vara mycket tidskrävande. Eftersom högdimensionella datamaterial inte går att visualisera, är det svårt att på förhand

intuitivt uppskatta vilka parametervärden som skapar en beslutsgräns som presterar väl. Detta problem löses genom att träna datamaterialet upprepade gånger med olika värden på parametrarna. (Abe, 2010, ss. 93, 97-99). Vid stora datamaterial ställs även krav på datorutrustning eftersom träningstiden är lång och kräver mycket minne (Abe, 2010, ss. 59-60).

På grund av flexibiliteten i parametrarna, vid applicering av en kärnfunktion, finns risk för överanpassning om felklassificeringsgraden skall hållas låg. Detta leder till en sämre generaliseringsförmåga (Steinwart & Christmann, 2008, s. 152). Metoder för hur modeller kan anpassas introduceras i Kapitel 5.

3 Teori bakom beslutsgränsen

Detta kapitel beskriver hur SVM anpassas vid de enklaste förutsättningarna. Detta innebär att datamaterialets två klasser är perfekt linjärt separabla, likt datamaterialet i Figur 2.1. Först ges en beskrivning av hur det optimala hyperplanet och marginalbredden är matematiskt definierade. Därefter förklaras hur marginalen maximeras.

3.1 Det optimala hyperplanet

För att kunna räkna ut det optimala hyperplanet krävs förutsättningen att datamaterialet går att separera linjärt. Utgångspunkten är att träningsmaterialet utgörs av n stycken observationer och består av p dimensioner. Varje observation benämns $\mathbf{x}_i$ där $i = 1, \dots, n$ och har en tillhörande klass $y_i = (-1,1)$ beroende på vilken klass observationen tillhör (Abe, 2010, s. 21). Sammanfattningsvis kan träningsmaterialet, D, skrivas som

$$D = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{-1,1\}\}$$

där $i = 1, \dots, n$.

Det optimala hyperplanet, H_0, bestäms av normalvektorn $\mathbf{w}$ som är vinkelrät mot hyperplanet och definieras av ekvationen

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

I det tvådimensionella fallet (dvs $p = 2$) skrivs vektorerna som $\mathbf{w} = (-a, 1)$ och $\mathbf{x} = (x_1, x_2)$ vilket gör hyperplanet likvärdigt med den räta linjen $x_2 = ax_1 - b$. Omskrivningen till vektorform görs av praktiska skäl, då denna form kan generaliseras till fler dimensioner än två.

Målet med SVM är att maximera avståndet mellan klasserna, det vill säga marginalbredden, $2m$. Att hitta det optimala hyperplanet blir således detsamma som att maximera marginalen, m (Hamel, 2009, s. 77).

3.1.1 Optimeringsproblemet

Hyperplanet är skalinvariant. Detta innebär att det är möjligt att multiplicera ekvationen $\mathbf{w}^T\mathbf{x} + b$ med valfri positiv konstant utan att hyperplanet påverkas (Abu-Mostafa et al., 2012b, s. 4). Denna frihet gör det möjligt att införa bivillkoret

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \tag{3.1}$$

och om $\mathbf{x}_i$ är en stödvektor och tillhör H_+ eller H_- så är

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

I praktiken innebär bivillkoret att det optimala hyperplanet inte kan väljas så att det befinner sig utanför H_+ eller H_-. Detta skulle inte uppfylla bivillkoret i Ekvation 3.1 för alla observationer.

Med hjälp av vektorn **w**, som är vinkelrät mot hyperplanet, och valfri stödvektor går det med hjälp av ortogonal projektion visa att marginalen är

$$m = \frac{1}{\|\mathbf{w}\|}.$$

Ortogonal projektion beskrivs närmare i Bilaga 1.

Att maximera marginalen är följaktligen detsamma som att minimera längden av **w**. För att underlätta framtida beräkningar, skrivs detta optimeringsproblem om till

$$\min \frac{1}{2}\|\mathbf{w}\|^2$$

under tidigare bivillkor

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

för alla *i* = 1, ..., *n* (Abu-Mostafa et al., 2012b, ss. 5-7).

3.1.2 Lagrange-multiplikatorer

Optimeringsproblem under bivillkor kan lösas genom att införa Lagrange-multiplikatorer. Dessa används för att hitta de lokala maximi- och minimipunkterna av en differentierbar funktion med tillhörande likhets- eller olikhetsbivillkor (Bishop, 2006).

Definitionsmässigt innebär detta att en funktion $f(x_1, \dots, x_n) : \mathbb{R}^n \to \mathbb{R}$ optimeras under bivillkoren $g(x_1, \dots, x_n) = 0$. Grunden till Lagrange-funktioner utgår från att gradienten av f är lika med Lagrange-multiplikatorn, α_i, multiplicerat med gradienten av ett bivillkor g. Detta kan skrivas om till Lagrange-funktionen

$$L(x, \alpha) = f(x) - \alpha g(x).$$

För att hitta extrempunkterna sätts $\nabla L(x, \alpha) = 0$ (Smith, 2004).

När det optimala hyperplanet inom SVM skall beräknas gäller följande funktion f och bivillkor g_i

$$f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$$

och

$$g_i(\mathbf{w}, b) = y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \geq 0.$$

Funktionen och de n bivillkoren kombineras tillsammans med Lagrange-multiplikatorerna (Vapnik, 1995, ss. 129-130)

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\alpha}) &= f(\mathbf{w}) - \sum_{i=1}^{n} \alpha_i g_i(\mathbf{w}, b) \\ &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i y_i[(\mathbf{w}^T\mathbf{x}_i + b) - 1]. \end{aligned} \tag{3.2}$$

Gradienten av Lagrange-funktionen beräknas genom

$$\nabla L(\mathbf{w}, b, \alpha) = \nabla f(\mathbf{w}) - \sum_{i=1}^{n} \alpha_i \nabla g_i(\mathbf{w}, b)\,,$$

vilket i detta fall blir

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = \mathbf{0} \rightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{3.3}$$

och

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0 \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{3.4}$$

Ekvation 3.3 och Ekvation 3.4 substitueras in i den ursprungliga Lagrange-funktionen definierad i Ekvation 3.2 vilket ger

$$L(\alpha) = \frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i\right) \cdot \left(\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j\right) - \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i\right) \cdot \left(\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j\right) - b\sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i$$
$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j. \quad (3.5)$$

Slutligen maximeras Ekvation 3.5 med avseende på α under bivillkoren

$$\alpha_i \geq 0$$

och

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

Svårigheter med att lösa problemet analytiskt uppstår redan vid ett litet antal observationer och ekvationerna löses därför numeriskt med hjälp av kvadratisk programmering. Lösningen ges i form av vektorn $\boldsymbol{\alpha} = \alpha_1, \alpha_2, \ldots, \alpha_n$ (Vapnik, 1995, s. 131). Lagrange-multiplikatorerna, α_i, kan tolkas som hur stor påverkan stödvektor i har för beslutsgränsen, vilket innebär hur mycket en förflyttning eller borttagning av stödvektor i påverkar det optimala hyperplanet. Detta blir intuitivt när lösningen av $\mathbf{w}$ i Ekvation 3.3 analyseras. Eftersom $\alpha_i y_i \mathbf{x}_i = 0$ för de i där $\alpha_i = 0$, innebär detta att dessa observationer inte påverkar beräkningen av $\mathbf{w}$. Slutsatsen blir att Ekvation 3.3 kan förenklas till

$$\mathbf{w} = \sum_{\text{enbart stödvektorer } \mathbf{x}_i} \alpha_i y_i x_i. \quad (3.6)$$

Konstanten b bestäms sedan genom att lösas ut ur uttrycket

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

för valfri stödvektor.

När sålunda alla okända parametrar skattats kan det optimala hyperplanet bestämmas av de **x** för vilka

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

Det optimala hyperplanet utgör även grund för den beslutsfunktion som används när nya observationer ska predikteras. Genom att substituera **w** enligt Ekvation 3.6 ges följande funktion (Vapnik, 1995, s. 131)

$$f(\mathbf{x}) = \sum_{\text{enbart stödvektorer } \mathbf{x}_i} \alpha_i y_i \left(\mathbf{x}_i^T \mathbf{x}\right) + b\,. \tag{3.7}$$

I de fall där $f(\mathbf{x}) > 0$, predikteras en observation tillhöra klass $y = 1$ och i de fall där $f(\mathbf{x}) < 0$, predikteras en observation tillhöra klass $y = -1$.

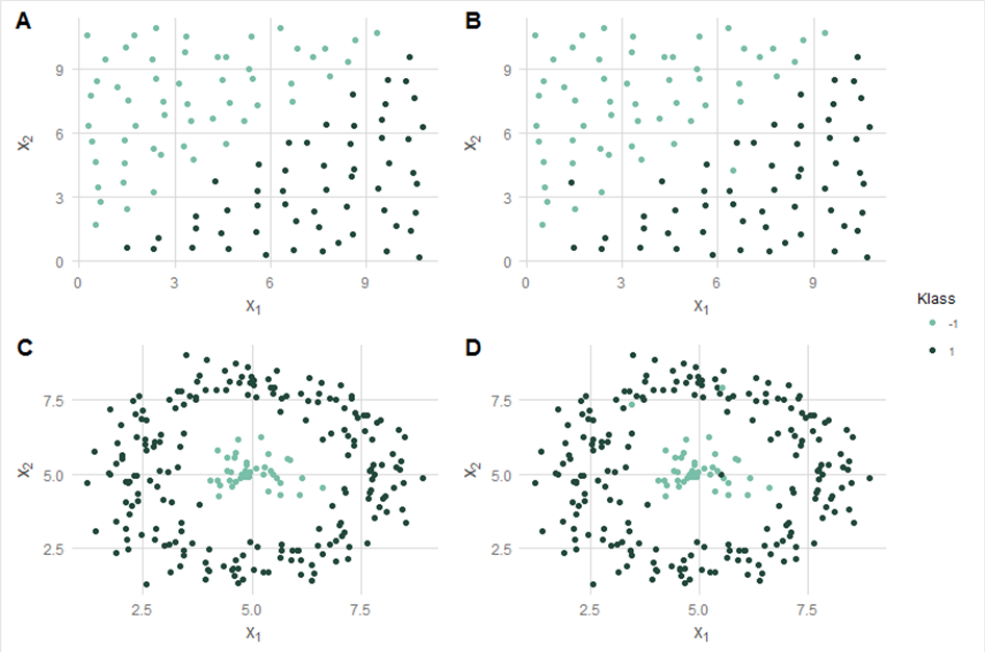
4 Utveckling av grundmodellen

I detta kapitel introduceras två metoder som kan användas för att utveckla grundmodellen till att lösa klassificeringsproblem när datamaterialets klasser inte är linjärt separabla. Först illustreras icke-linjära exempel. Vidare beskrivs mjuk marginal samt kärnfunktioner.

4.1 Icke-linjära exempel

I exempel från tidigare kapitel har datamaterialets klasser både varit linjärt separabla och fria från *outliers*. I mer realistiska scenarion uppfyller datamaterialet sällan dessa egenskaper fullständigt. Det går då att utveckla modellen till att dels separera datamaterialet icke-linjärt och dels bortse från enstaka outliers. Nedan redogörs för hur klasser separeras i dessa scenarion, samt vilken typ av icke-linjäritet som ger upphov till vilken typ av modellutveckling.

I Figur 4.1 illustreras enkla exempel av de fyra kombinationerna av dessa två egenskaperna i två dimensioner. I Figur 4.1A är klasserna linjärt separabla utan outliers. I Figur 4.1B är klasserna linjärt separabla bortsett från några enstaka



Figur 4.1 *Exempel på olika klasstrukturer i ett datamaterial. I Figur A är datamaterialet linjärt separabelt. I Figur B är datamaterialet generellt linjärt separabelt bortsett från enstaka outliers. I Figur C är datamaterialet icke-linjärt separabelt. I Figur D är datamaterialet icke-linjärt separabelt, med enstaka outliers.*

outliers. I Figur 4.1C är klasserna icke-linjärt separabla utan enstaka outliers. I Figur 4.1D är klasserna icke-linjärt separabla med ett fåtal förekommande outliers.

4.2 Mjuk marginal

I situationer där klasser är linjärt separabla men enstaka outliers förekommer, är det sällan rekommenderat att anpassa en ickelinjär beslutsgräns (se Kapitel 4.3). Detta kan leda till att modellen överanpassas. Det är istället önskvärt att bortse från de avvikande observationerna när det optimala hyperplanet beräknas (Abu-Mostafa et al., 2012b, ss. 40-41).

Detta löses genom att införa en så kallad mjuk marginal (”soft margin” på engelska). När modellen anpassas med en mjuk marginal tillåts observationer befinna sig innanför marginalen mellan H_+ och H_-. Observationer tillåts dessutom befinna sig på fel sida av det optimala hyperplanet. Modellen skiljer sig inte markant från grundmodellen (Vapnik, 1995, s. 133). I detta delkapitel ges en kort beskrivning av de skillnader som finns mellan grundmodellen och en modell med mjuk marginal.

I Kapitel 3 begränsades det optimala hyperplanets anpassning av bivillkoret $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$. Med en mjuk marginal ersätts detta bivillkor med

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

där $\xi_i \geq 0$ och representerar hur mycket varje observation bryter mot bivillkoret $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$. För de punkter som befinner sig utanför marginalen mellan H_+ och H_- är $\xi_i = 0$, medan ξ_i växer desto större överträdelsen blir. Felmåttet för modellen definieras som

$$\sum_{i=1}^{n} \xi_i$$

vilket motsvarar den totala överträdelsen av observationerna.

Detta leder till att det tidigare optimeringsproblemet

$$\min \frac{1}{2}\|\mathbf{w}\|^2$$

omformuleras till

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + c\sum_{i=1}^{n}\xi_i$$

under bivillkor

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

och

$$\xi_i \geq 0$$

för alla $i = 1, ..., n$.

Här är c, felkostnaden, en justerbar parameter som representerar graden av överträdelser som tillåts. Ett stort värde på c tillåter få observationer att ligga på fel sida om H_+ eller H_- vid anpassning av det optimala hyperplanet. Ett litet värde på c tillåter många observationer att överträda H_+ eller H_- (Hastie, Tibshirani, & Friedman, 2009, s. 424).

Lagrange-multiplikatorerna, α_i, beräknas nästintill identiskt med hur de beräknades i Ekvation 3.5. Skillnaden är att beräkningen begränsas av bivillkoret (Abu-Mostafa et al., 2012b, ss. 41-43)

$$0 \leq \alpha_i \leq c$$

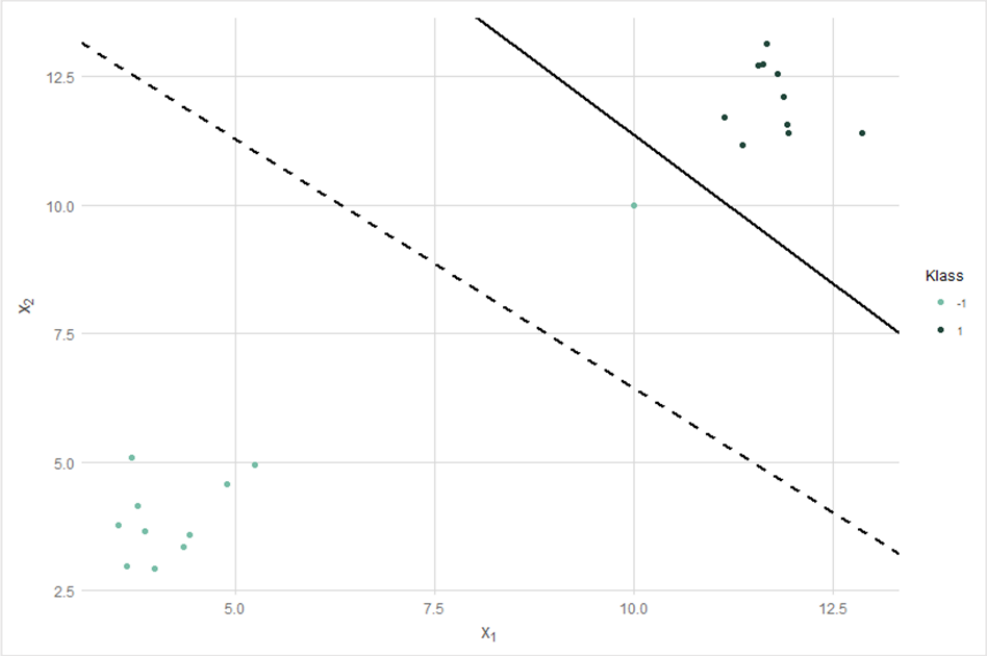
istället för det tidigare bivillkoret

$$0 \leq \alpha_i \leq \infty.$$

Slutligen introduceras träningsfelet

$$E_{IN} = \frac{1}{n}\sum_{i=1}^{n} e_i$$

där e_i antar värdet 1 ifall observation i befinner sig på fel sida av H_+ eller H_-, och värdet 0 ifall en observation befinner sig på rätt sida av H_+ och H_-. Detta mått mäter andelen överträdelser vid modellens anpassning. På grund av tidigare nämnda anledningar så innebär inte $E_{IN} > 0$ att modellen är obrukbar, utan det kan vara direkt önskvärt att uppnå en viss grad av E_{IN} för att inte överanpassa modellen.



Figur 4.3 *Illustration av skillnaden mellan grundmodellen och mjuk marginal. Den streckade linjen är den mjuka marginalen. Den heldragna linjen är grundmodellens marginal. Detta är ett exempel på en situation där det kan vara önskvärt att bortse från outliern vid modellanpassningen för att få ett mer generaliserbart hyperplan.*

Ett exempel på detta illustreras i Figur 4.2. Här ligger en observation med klasstillhörighet -1 väldigt nära observationer med klasstillhörighet 1. I övrigt förekommer stor skillnad mellan klasserna. Trots att klasserna är fullt linjärt separabla, kan $E_{IN} \neq 0$ och anpassning av det optimala hyperplanet med mjuk marginal vara önskvärt i detta fall (streckad linje). Om däremot denna outlier inte bortses från, kommer den att få stort inflytande vid anpassningen av hyperplanet (heldragen linje).

4.3 Kärnfunktioner

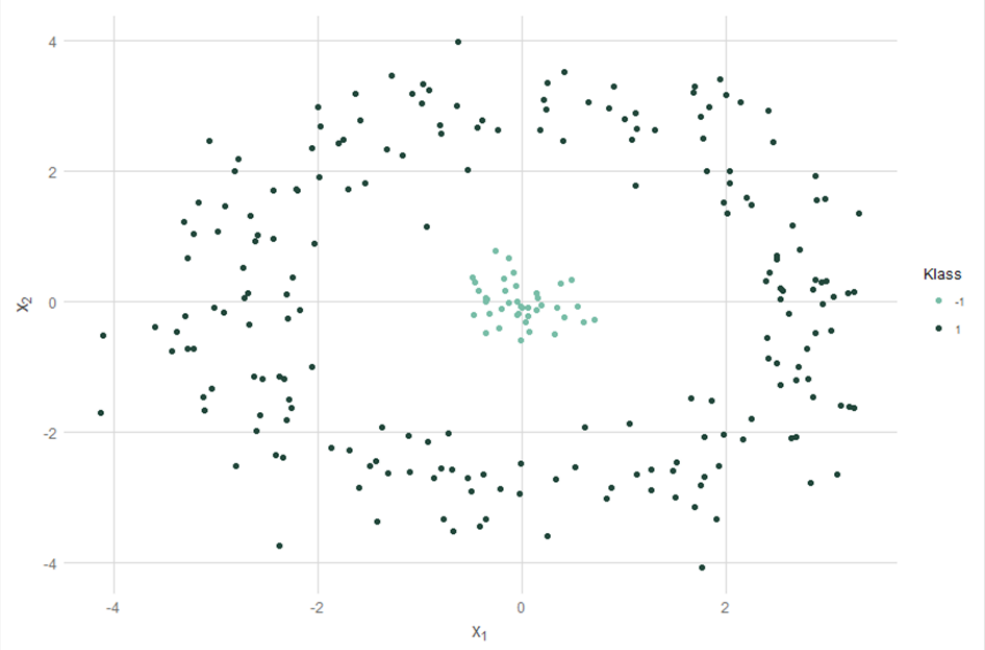
Det finns många situationer där det inte går att separera datamaterialets klasser linjärt. I Figur 4.3 är det exempelvis olämpligt att anpassa en modell med en linjär beslutsgräns även om överträdelser tillåts genom att applicera en mjuk marginal. Inom SVM hanteras denna situation genom att applicera en kärnfunktion (”kernel” på engelska) som utökar det ursprungliga rummet till ett nytt rum av högre dimension (James, Witten, Hastie, & Tibshirani, 2013, ss. 349-350). Detta kan ge möjlighet att separera klasserna linjärt i det nya rummet.

Figur 4.3 illustrerar en situation där det är klart olämpligt att separera datamaterialets klasser linjärt. Oavsett var beslutsgränsen placeras blir generaliseringsförmågan hos

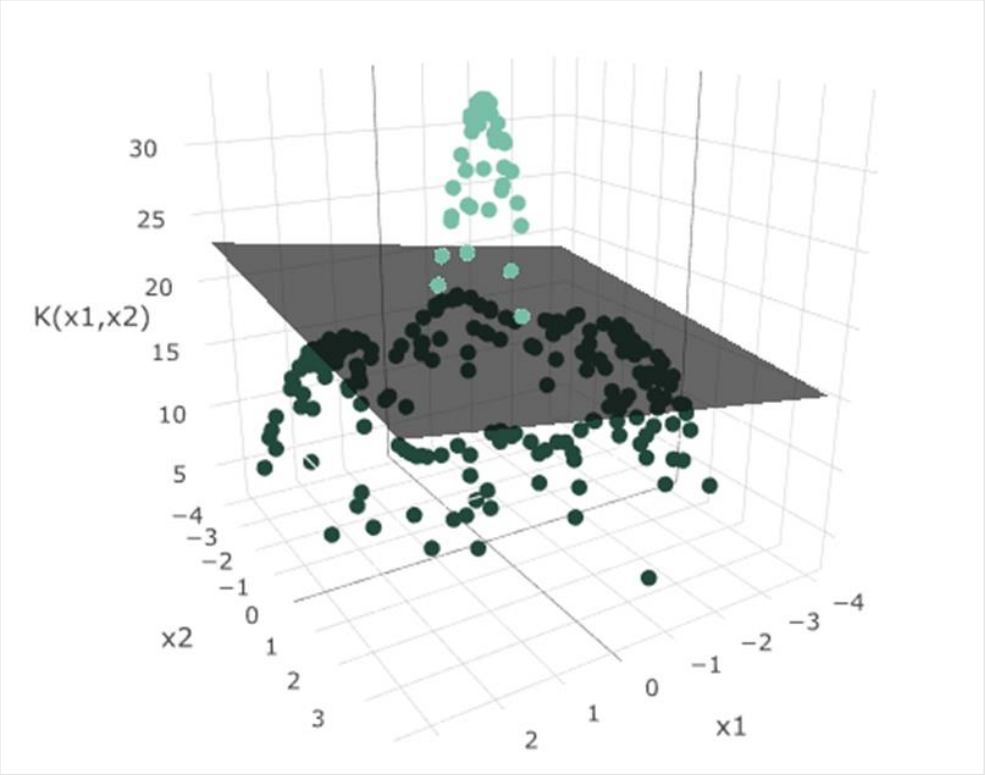
modellen svag. I Figur 4.4 har ytterligare en dimension införts, vilket gör det möjligt att linjärt separera materialet. Det linjära hyperplan som separerar datamaterialet i det utökade rummet motsvaras av en ickelinjär beslutsgräns i det ursprungliga rummet. Den icke-linjära beslutsgränsen illustreras i Figur 4.5.

En transformation av det ursprungliga rummet kan se ut på många olika sätt. Rummet skulle kunna låtas anta ett väldigt stort antal dimensioner, men transformationen blir snabbt beräkningsmässigt svårhanterlig (Gareth et al., 2013, s. 350). Med hjälp av applicering av en kärnfunktion går det att anpassa en linjär beslutsgräns i ett högdimensionellt rum utan att bli beräkningsmässigt bestraffade (Gareth et al., 2013, s. 351). Detta beror på att det optimala hyperplanet och beslutsfunktionen endast beror på den inre produkten av $\mathbf{x}^T\mathbf{x}$ (se Ekvation 3.5 och Ekvation 3.7), och aldrig $\mathbf{x}$ enskilt.

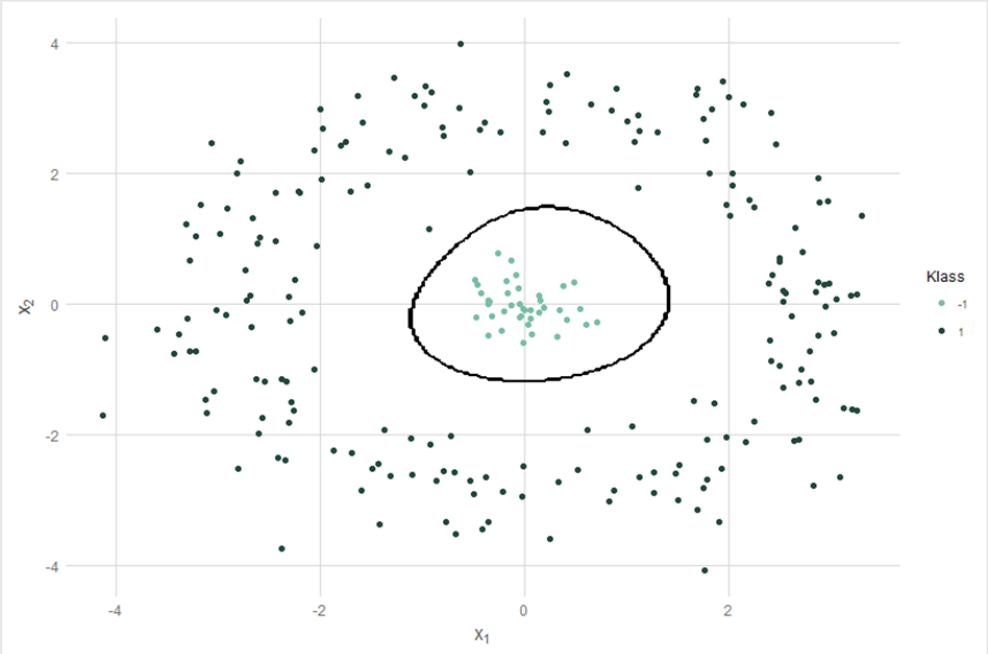
En kärnfunktion $K(\mathbf{x}_i, \mathbf{x}_j)$ kan utvecklas till en inre produkt av $\mathbf{x}_i^T\mathbf{x}_j$ i en högre dimension. Därmed behöver inte rummet av högre dimension explicit hanteras (Abe, 2010, s. 32). Ett exempel på detta demonstreras i Bilaga 2.



Figur 4.4 *Exempel på ett datamaterial med två tydliga klasser som inte är linjärt separabla. I denna situation är det olämpligt att anpassa en linjär beslutsgräns med mjuk marginal. Detta är en typisk situation där applicering av en kärnfunktion behövs.*



Figur 4.4 *Illustration av en kärnfunktion-applicering på Z-axeln som skapar möjligheten att linjärt separera klasserna i det utökade rummet.*



Figur 4.5. *I det ursprungliga rummet representeras det optimala hyperplanet i Figur 4.4 av en icke-linjär beslutsgräns.*

Med utgångspunkt i den tidigare definierade beslutsfunktionen i Ekvation 3.7

$$f(\mathbf{x}) = \sum_{\text{enbart stödvektorer } \mathbf{x}_i} \alpha_i y_i \left(\mathbf{x}_i^T \mathbf{x}\right) + b$$

införs en kärnfunktion som ersätter den inre produkten $\mathbf{x}_i^T \mathbf{x}$. Beslutsgränsen skrivs i generell form således om till (Vapnik, 1995, s. 136)

$$f(\mathbf{x}) = \sum_{\text{enbart stödvektorer } \mathbf{x}_i} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b.$$

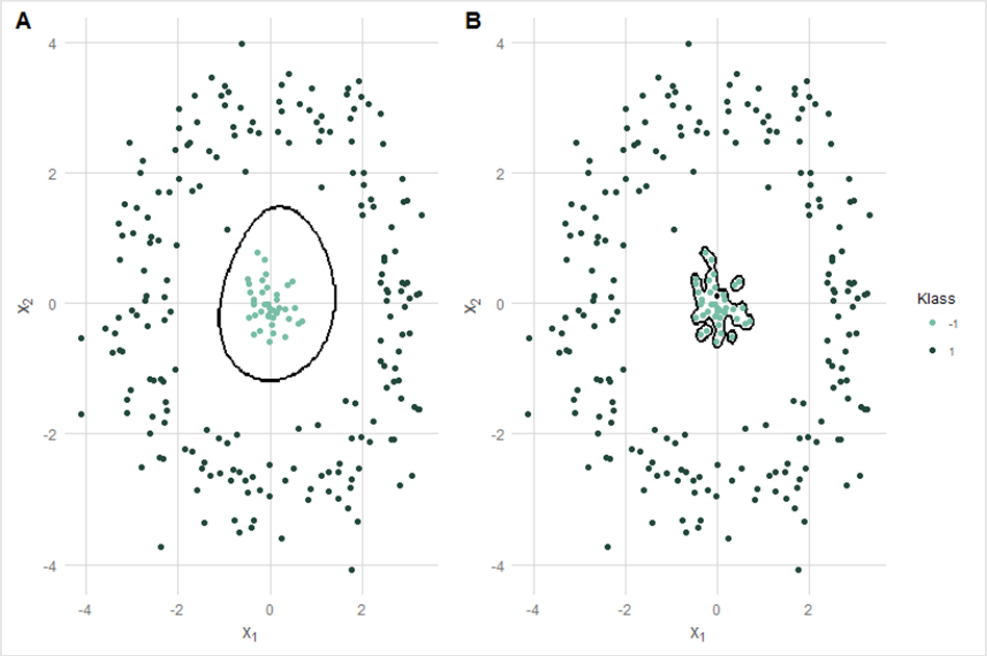
Några vanliga typer av kärnfunktioner återges i Tabell 4.1.

Tabell 4.1 *Beskrivning av de vanligaste typerna av kärnfunktioner inom SVM.*

Namn	Funktion
Linjär	$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)$
Polynom	$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^d$
Gaussisk	$K(\mathbf{x}_i, \mathbf{x}_j) = exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)$

En linjär kärnfunktion separerar datamaterialets klasser linjärt i det ursprungliga rummet. Den är därmed identisk med grundmodellen som introducerats i Kapitel 3. Mer avancerade kärnfunktioner har ytterligare parametrar som kan justeras för att forma beslutsgränsen. Exempelvis använder den Gaussiska kärnfunktionen parametern γ som justerar beslutsregionen efter hur varje enskild observation påverkar. Detta visualiseras i Figur 4.6. I Figur 4.6A har en Gaussisk kärnfunktion anpassats till datamaterialet med $\gamma = 0.5$. I Figur 4.6B har samma kärnfunktion anpassats med $\gamma = 100$. I Figur 4.6B är beslutsgränsen tydligt överanpassad.

Komplexiteten hos modellen kan anpassas och styras genom att välja typ av kärnfunktion och dess tillhörande parametervärden. (Hamel, 2009, s. 107). Det är ofta svårt att visuellt undersöka vilken typ av kärnfunktion som behövs för att kunna separera ett datamaterial. Består materialet dessutom av fler än tre attribut går det inte att visualisera alls. Kapitel 5.3 behandlar hur parametervärden väljs i praktiken.



Figur 4.6 *Illustrering av hur värdet på γ påverkar beslutsgränsen. I Figur A är γ=0.5. I Figur B är γ=100. I Figur B är beslutsgränsen tydligt överanpassad.*

5 Anpassning av modell och prestandamått

I detta kapitel introduceras utvalda mått som används vid anpassning och val av modell. Först introduceras generella felklassificeringsmått, följt av mer specifika mått som berör modellens träffsäkerhet. Slutligen ges exempel på tillvägagångssätt vid modellval.

5.1 Modellvalidering

Ett viktigt prestandamått vid modellvalidering är att skatta *out of sample-felet*, E_{OUT}. Out of sample-felet mäter generaliseringsförmågan hos den anpassade modellen. Två förslag på vanligt förekommande metoder för att skatta E_{OUT} är genom *utelämna-en-korsvalidering* eller genom att dela upp datamaterialet i en *träningsmängd* och en *testmängd*.

Utelämna-en-korsvalidering innebär att en observation i taget utelämnas vid modellanpassning för att sedan klassificeras. Vid utelämna-en-korsvalidering beräknas

$$E_{CV} = \frac{1}{n}\sum_{i=1}^{n} e_i$$

där E_{CV} står för utelämna-en-korsvalideringsfelet och e_i hur observation i har klassificerats. Klassificeras observation i korrekt, antar e_i värdet 0. Skulle observation i klassificeras felaktigt, antar e_i värdet 1. Eftersom endast stödvektorer påverkar hyperplanet kan inte en observation som inte är stödvektor felklassificeras vid en utelämna-en-korsvalidering. Dessa observationer befinner sig per definition på rätt sida av stödvektorerna (Abu-Mostafa et al., 2012b, s. 18). Således gäller

$$E_{CV} = \frac{1}{n}\sum_{i=1}^{n} e_i \leq \frac{\#\ \text{stödvektorer}}{n}.$$

E_{CV} är en väntevärdesriktig skattning av out of sample-felet, E_{OUT}, med $n-1$ observationer.

Det förväntade out of sample-felet beror endast på andelen stödvektorer i datamaterialet (Abu-Mostafa et al., 2012b, s. 18). Andelen stödvektorer ger därmed en fingervisning om hur generaliserbar modellen är.

Vidare kan E_{OUT} skattas genom att dela upp datamaterialet i en träningsmängd och en testmängd. Träningsmängden används vid anpassning av modellen. Modellen valideras sedan genom att prediktera testmängdens observationer på modellen (Abu-Mostafa et al., 2012a, ss. 61, 138). Detta mått beräknas som

$$E_{TEST} = \frac{1}{n'} \sum_{i'=1}^{n'} e_{i'}$$

där n' står för storleken på testsmängden och $e_{i'}$ hur observation i' klassificeras. Även E_{TEST} är en väntevärdesriktig skattare av E_{OUT}.

Den rekommenderade kvoten mellan träningsmängd och testmängd varierar. Vanliga förslag på kvoter är mellan 90–10 och 70–30 beroende på datamaterialets storlek. Mindre datamaterial kräver en högre andel observationer i testmängden (Hagan, Demuth, Beale, & De Jesús, 2014; Guyon, u.å.).

5.2 Träffsäkerhet

Vid binär klassificering benämns de två klasserna generellt som *positiva* och *negativa*. Dessa begrepp har nödvändigtvis ingen praktisk koppling till de två klasserna, utan kan liknas vid en binär kategorivariabel som antar värdet 0 eller 1. I detta kapitel kommer dessa klasser hänvisas till som positiv och negativ.

Det finns fyra möjliga utfall när en observation klassificeras. Dessa illustreras i Tabell 5.1.

Tabell 5.1 *Tabell över de fyra möjliga utfall vid binär klassificering. Denna typ av tabell benämns confusion matrix.*

	Faktiska negativ	Faktiska positiv
Predikterade negativ	Sanna negativ	Falska negativ
Predikterade positiv	Falska positiv	Sanna positiv

Det mest generella måttet på modellens prestanda ges av felklassificeringsgraden.

$$Felklassificeringsgrad = 1 - \frac{Sanna\ positiv + Sanna\ negativ}{n}$$

Ett flertal andra kvoter baserade på Tabell 5.1 mäter mer specifika förmågor hos modellen. Dessa kan vara av varierande intresse beroende på syftet med analysen. Nedan redogörs för tre av dessa. Två vanliga mått är modellens sensitivitet och

specificitet. Sensitivitet mäter hur väl modellen klassificerar observationer med positiv klasstillhörighet och definieras som

$$Sensitivitet = \frac{Sanna\ positiv}{Faktiska\ positiv}.$$

Specificitet mäter hur väl modellen klassificerar observationer med negativ klasstillhörighet och definieras som (Ledolter, 2013, s. 109)

$$Specificitet = \frac{Sanna\ negativ}{Faktiska\ negativ}.$$

Ett mått som inte fångas av sensitiviteten och specificiteten är modellens precision. Precision definieras som

$$Precision = \frac{Sanna\ positiv}{Predikterade\ positiv}.$$

Likt modellens sensitivitet så bidrar modellens precision med information om de observationer som klassificerats som positiva. Skillnaden utgörs av att precisionen mäter träffsäkerheten i de fall modellen klassificerar en observation som positiv (Powers, 2011).

5.3 Grid search och val av modell

I de fall datamaterialet består av fler dimensioner än tre går det inte på förhand bedöma om det finns ett behov av en mjuk marginal, kärnfunktion eller en kombination av de två. Detta gäller även för val av tillhörande parametervärden.

Grid search är en rekommenderad metod för att hitta bästa möjliga modell (Abe, 2010, ss. 97-99). En grid search genomförs genom att välja ut ett förutbestämt antal värden som parametrarna får anta. För att fånga både väldigt låga och väldigt höga värden rekommenderas att välja värden efter olika potenser av någon bas, det vill säga

$$\boldsymbol{k_j} = (k_j^i, k_j^{i+1}, \dots, k_j^{i+a})$$

där k_j är en konstant och exponenten i varierar mellan i och $i + a$.

Modeller anpassas för varje kombination av $k_1, \dots, k_p$, där p är antalet parametrar den valda kärnfunktionen använder (Hsu, Chang, & Lin, 2003). Beroende på syftet

med analysen, väljs en av dessa modeller ut med utgångspunkt i ett eller flera av de mått som introducerades i Kapitel 5.1 och 5.2.

6 R-paketet e1071

I R-paketet *e1071* finns det möjlighet att skapa en SVM-modell med hjälp av funktionen *svm()*. Kärnfunktionstyp bestäms av parametern *kernel*. De vanliga kärnfunktionerna som tidigare introducerats väljs här genom att sätta parametern *kernel* till ”linear”, ”polynomial” eller ”radial”, där ”radial” motsvarar den Gaussiska kärnfunktionen.

Modellparametrarna γ, d och c bestäms av parametrarna *gamma*, *degree* och *cost*. Parametern *scale* accepterar värdena TRUE eller FALSE. Denna parameter beslutar om variablerna ska standardiseras/skalas eller inte, där TRUE är default.

SVM kan användas både till klassificering och regression. Genom att ändra parametern *type* manuellt är det är möjligt att välja vilken typ av SVM som ska anpassas. Funktionen väljer *type* automatiskt beroende på om y är en kategorivariabel (*factor*) eller numerisk.

För mer information om R-paketet *e1071* se *help(e1071)* eller https://cran.r-project.org/web/packages/e1071/e1071.pdf.

7 Datamaterialet Wisconsin Diagnostic Breast Cancer

I detta kapitel presenteras det datamaterial som användas i analysen. Inledningsvis ges en beskrivning av materialet, följt av deskriptiv statistik och hur materialet har redigerats inför analysen.

7.1 Beskrivning av datamaterial

Det datamaterial som används i analysen är Wisconsin Diagnostic Breast Cancer som är skapat av Dr. William H. Wolberg, Nick Street och Olvi L. Mangasarian. Materialet är är ursprungligen insamlat genom biopsier av Dr. William Wolberg vid Wisconsins universitetssjukhus mellan åren 1989 och 1991. Dessa tester har sedan bildanalyserats och givit grunden för de mätvärden som är registrerade i datamaterialet. Datamaterialet är hämtat från UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic).

Tabell 7.1 *Beskrivning av variabler i datamaterialet.*

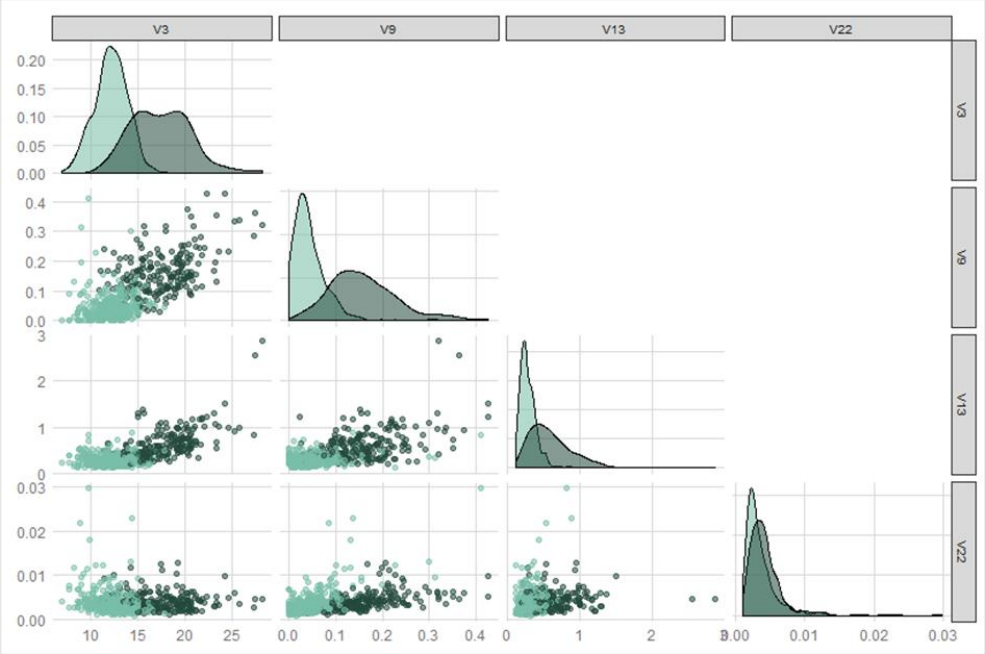
Variabelkod	Variabeletikett	Beskrivning	Datanivå
V3, V13, V23	Radie	Medelvärdet av avståndet från mittpunkten till punkter på omkretsen	Numerisk
V4, V14, V24	Textur	Standardavvikelse av gråskalvärden	Numerisk
V5, V15, V25	Omkrets	Omkrets av cellkärnan	Numerisk
V6, V16, V26	Area	Area av cellkärnan	Numerisk
V7, V17, V27	Släthet	Lokala variationer i radien	Numerisk
V8, V18, V28	Kompakthet	Omkrets ^{2}/Area – 1	Numerisk
V9, V19, V29	Konkavitet	Konkavitetsgrad	Numerisk
V10, V20, V30	Konkava punkter	Andelen konkava delar av konturen	Numerisk
V11, V21, V22	Symmetri	Cellkärnans symmetri	Numerisk
V12, V22, V32	Fraktaldimension	”Kustlängdsapproximation” – 1	Numerisk
V2	Klass	Klasstillhörighet (Godartad, elakartad)	Kategori

Datamaterialet innehåller 32 olika variabler och 569 observationer. Variabel 1 är ett ID-nummer och variabel 2 representerar klasstillhörighet (godartad eller elakartad). Variabel 3–32 mäter medelvärde, standardavvikelse samt högsta värde (medelvärdet av de tre högsta mätvärdena) för tio olika cellegenskaper. Exempelvis motsvarar variabel 3, 13 och 23 olika mätvärden av cellkärnans radie. De tio egenskaperna beskrivs i Tabell 7.1.

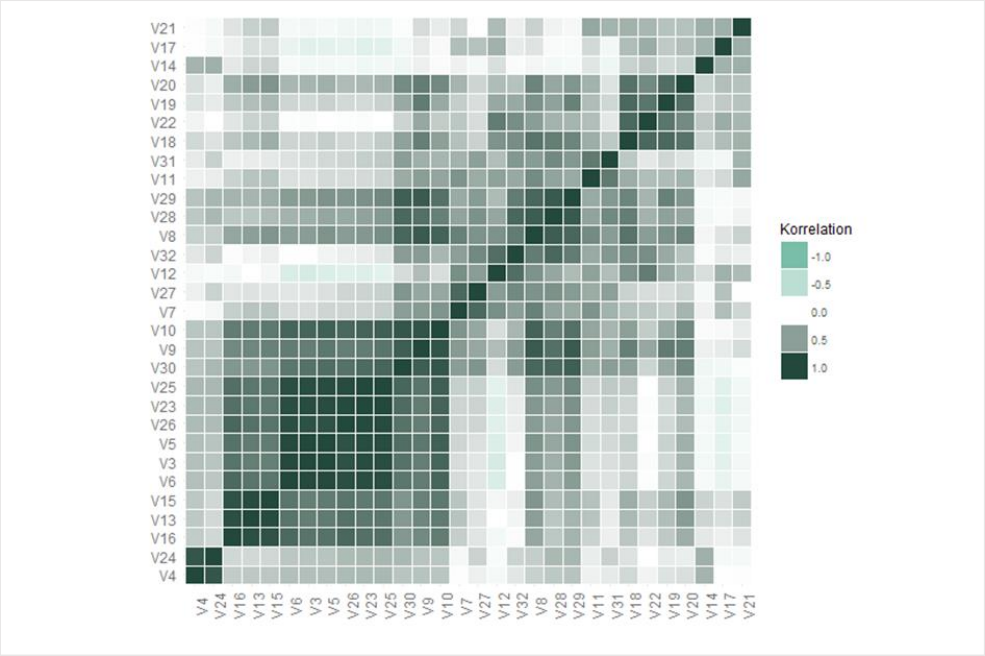
7.2 Deskriptiv statistik

Klassvariabeln, V2, är fördelad med 357 (62.7 %) godartade observationer och 212 (37.3 %) elakartade observationer.

I Figur 7.1 illustreras densitet och punktdiagram för fyra utvalda variabler. Diagram över samtliga variabelkombinationer redovisas ej här på grund av utrymmesmässiga skäl. Syftet med diagrammen är att få en överblick över datamaterialets struktur. Det går att utläsa tydliga skillnader mellan de två klasserna för majoriteten av variablerna. Undantagsfall är variabeln V22 som inte visar lika tydliga skillnader.



Figur 7.1 *Punktdiagram över kombinationer av V3, V9, V13 och V22. Det går i många fall att uttyda relativt stora skillnader mellan klasserna. Densitetdiagrammen i diagonalen visar tydliga klasskillnader för V3, V9 och V13.*



Figur 7.2 *Korrelationsmatris över datamaterialet. Hög korrelation förekommer mellan flera block av variabler.*

En korrelationsmatris illustreras i Figur 7.2. Flera block av variabler med hög korrelation förekommer.

7.3 Dataredigering

Variabel 1 (ID) har tagits bort då den inte är relevant för analysen. Övrigt material har slumpmässigt delats in i träningsmängd och testmängd. Träningsmängden består av 70% av observationerna och testmängden resterande 30%. Detta ger en träningsmängd på 398 observationer och en testmängd på 171 observationer. Eftersom variablerna varierar kraftigt i skala så standardiseras dessa vid modellanpassningen.

8 Analys

I detta kapitel används SVM för att anpassa modeller till datamaterialet Wisconsin Diagnostic Breast Cancer som presenterades i förra kapitlet. Först presenteras de anpassade modellerna. Vidare motiveras valen av de modeller som ska valideras på testmängden. Slutligen presenteras den slutgiltiga modellen, dess prestationsförmåga samt en kort undersökning av en observation som kan visas vara felklassificerad.

8.1 Träning av modeller

Datamaterialet, efter att klassindelningsvariabeln exkluderats, består av 30 dimensioner. Det går därför inte enkelt att grafiskt studera datastrukturen, och från denna anpassa lämpliga parametervärden. Grid search används därför för att hitta bästa möjliga modell.

Två grupper av modeller med mjuk marginal anpassas. I Modellgrupp I anpassas modeller med en linjär kärnfunktion och parametervärdet $c = 10^i$ där $i = -4, -3, \dots, 3$, och i Modellgrupp II anpassas modeller med en Gaussisk kärnfunktion och parametervärdena $c = 10^i$ där $i = -2, -1, \dots, 4$ och $\gamma = 10^j$ där $j = -5, -4, \dots, 0$. Den praktiska tolkningen av parametrarna c och γ har beskrivits i Kapitel 4.1 och 4.2.

Totalt skapas åtta modeller för modellgrupp I och 42 modeller för modellgrupp II. Vid modellskattningen korsvalideras samtliga modeller med utelämna-en-metoden. Den bästa modellen från varje modellgrupp väljs ut sedan för vidare analys. Korsvalideringens sensitivitet används främst som utgångspunkt vid detta val.

Tabell 8.1 *Beskrivning av de två modellgrupperna.*

	Marginal	Kärnfunktion	Parametrar
Modellgrupp I	Mjuk	Linjär	c
Modellgrupp II	Mjuk	Gaussisk	c, γ

Tabell 8.2 *Sammanställning av prestandamått för den bästa modellen i varje modellgrupp.*

	c	γ	Antal SV	E_{IN}	E_{CV}	$\text{Sensitivitet}_{\text{CV}}$
Modell 1	1	-	36	0.0151	0.02512	0.9602
Modell 2	10	0.01	53	0.0151	0.02512	0.9602

Figur 8.1 *Illustration av korsvaliderings-sensitiviteten givet olika kombinationer av parametervärden. Den högsta sensitiviteten hittas vid c=1 för den linjära modellgruppen (övre delen), och c=10 och γ=0.01 för den Gaussiska modellgruppen (undre delen). Dessa två modeller väljs ut för vidare analys.*

Eftersom syftet med analysen är att klassificera tumörer anses sensitivitet vara det viktigaste måttet. Detta är på grund av att det anses viktigare att fånga upp så många elakartade tumörer som möjligt än att minimera felklassificeringsgraden. Modellernas sensitivitet presenteras i Figur 8.1. Vidare kontrolleras att den valda modellen inte har ett extremt högt antal stödvektorer eller ovanligt högt totalt korsvalideringsfel, E_{CV}. De maximala sensitiviteterna i de två modellgrupperna är båda 96 %. De utvalda modellernas parametervärden och prestandamått redovisas i Tabell 8.2.

8.2 Validering med hjälp av en testmängd

De två utvalda modellerna från modellträningen används för prediktion av testmängdens observationer. Resultatet redovisas i Tabell 8.3 och 8.4.

Tabell 8.3 *Confusion matrix över prediktionen av testmängden för modell 1.*

Modell 1		
	Faktiskt godartad	Faktiskt elakartad
Predikterad godartad	109	2
Predikterad elakartad	1	59

Tabell 8.4 *Confusion matrix över prediktionen av testmängden för modell 2.*

Modell 2		
	Faktiskt godartad	Faktiskt elakartad
Predikterad godartad	109	1
Predikterad elakartad	1	60

Tabell 8.5 *Sammanställning av prestandamått vid modellvalidering för de två modellerna. Modell 2 uppnår en högre grad av sensitivitet. Denna väljs som slutgiltig modell.*

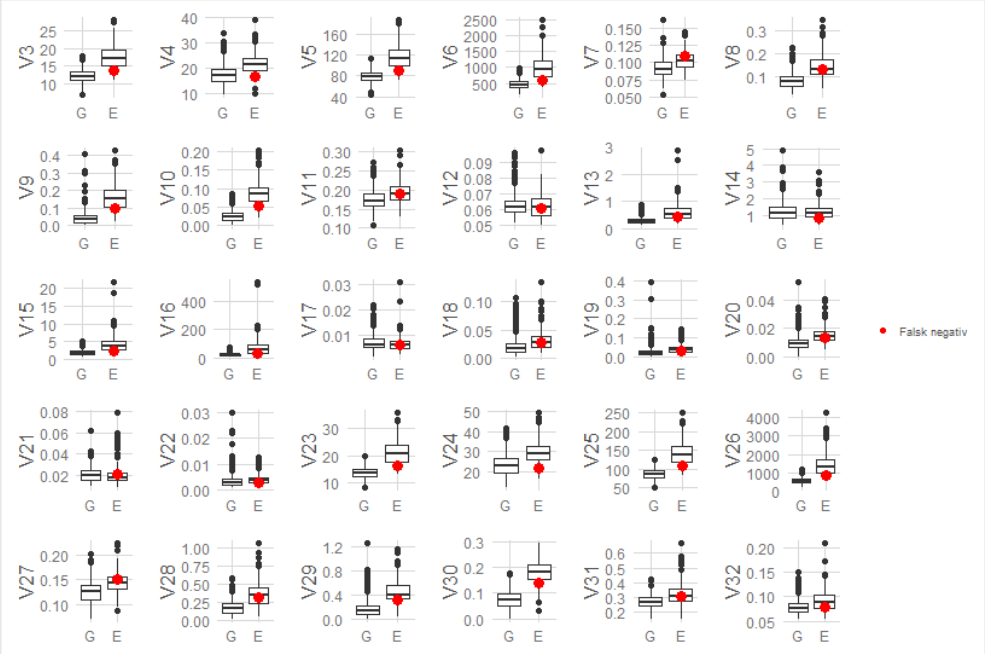
	Sensitivitet	Precision
Modell 1	0.9672	0.9833
Modell 2	0.9836	0.9836

I Tabell 8.5 presenteras relevanta mått på träffsäkerhet hos testmängden. Modell 2 har marginellt högre sensitivitet än Modell 1. Därmed väljs Modell 2 ut som slutmodell.

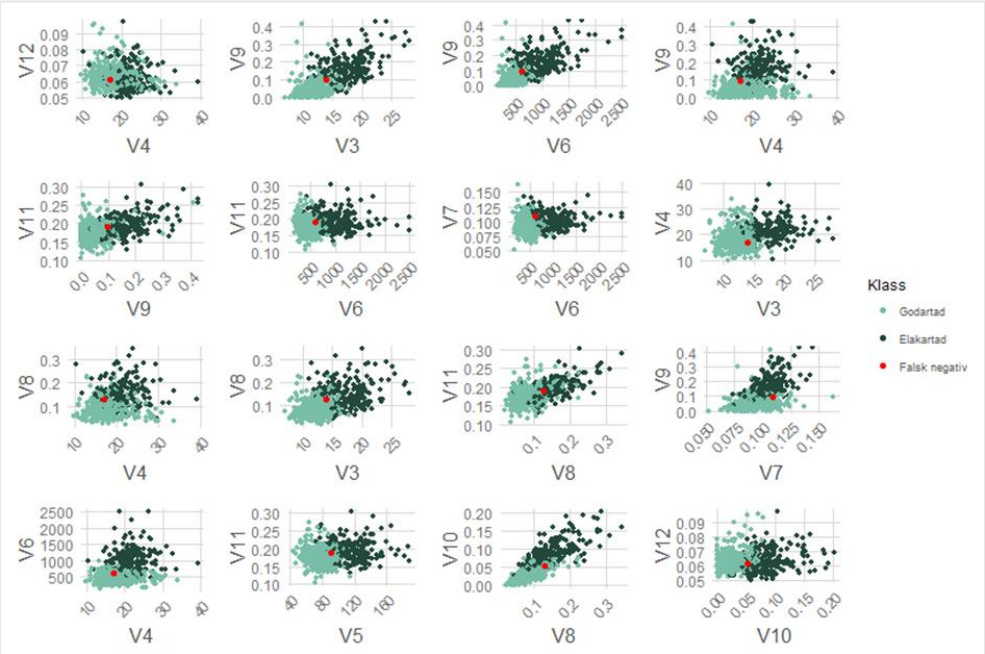
Eftersom stor vikt läggs på sensitivitet, undersöks mer ingående den observation från testmängden som felaktigt predikterats som godartad. Visualiseringar i ”boxplots” för samtliga 30 variabler uppdelat på klasstillhörighet (G=godartad, E=elakartad), presenteras i Figur 8.2. Den observation som felaktigt predikterats som godartad illustreras med en röd punkt. I flera fall befinner sig den falska negativa närmare den godartade klassens ”box” än den elakartade klassens.

Även punktdiagram över variabelkombinationer har undersökts. Exempel på utvalda kombinationer redovisas i Figur 8.3. De ljusgröna och mörkgröna punkterna illustrerar de två klasserna medan den röda punkten representerar den falska negativa.

I de flesta fall befinner sig den falska negativa på gränsen mellan de huvudkluster som bildats av de två klasserna. I vissa fall befinner sig observationen inom det område som domineras av godartade observationer. I ytterst få variabelkombinationer befinner sig observationen i en region som domineras av elakartade observationer.



Figur 8.2 *Boxplots över samtliga variabler uppdelat på klasstillhörighet. Den röda punkten representerar observationen som felaktigt klassificerats som godartad. I många fall befinner sig den felaktigt klassificerade observationen närmare boxen för den godartade klassen.*



Figur 8.3 *Punktdiagram över ett urval av variabelkombinationer. Den röda punkten representerar observationen som felaktigt klassificerats som godartad. I majoriteten av kombinationerna befinner sig observationen på gränsen mellan de två klasserna. Den befinner sig aldrig i ett område som domineras av elakartade observationer.*

9 Diskussion och slutsats

Denna uppsats ger en introduktion till SVM från teori till slutgiltig analys. Analysen ämnar ge en förståelse för hur metoden kan användas i praktiken och hur modeller kan anpassas efter bestämda önskemål med hjälp av olika prestandamått. Genom applicering av metoden på datamaterialet Wisconsin Diagnostic Breast Cancer anpassas modeller för att prediktera tumörer som elakartade eller godartade.

Den slutgiltiga modellen presterar i linje med förutbestämda önskemål. Syftet med analysen av datamaterialet är att fånga upp så många elakartade tumörer som möjligt och därmed är modellens sensitivitet av största intresse. Vid validering på testmängden är det enbart en elakartad tumör som inte klassificeras korrekt. Valideringen tyder även på att modellen har hög generaliserbarhet och således inte har överanpassats. Två huvudpunkter bedöms viktiga för vidare diskussion. Dessa berör dels den höga korrelation som finns mellan många av datamaterialets variabler och dels den observation som felaktigt klassificerats som godartad, den falska negativa.

Vid användning av en Gaussisk kärnfunktion, kan korrelation påverka den anpassade beslutsgränsen. Det ska dock påpekas att något modellantagande inte har brutits, eftersom några antaganden gällande korrelation inte har gjorts. Däremot är det ändå värt att notera att beslutsgränsen kan anta ett annat utseende om korrigering för korrelationen görs. Det går exempelvis att korrigera för korrelationen genom att kombinera vald kärnfunktion med *Mahalanobis distance*. Se exempelvis Bhavsar & Ganatra (2015) och Haasdonk & Pekalska (2009) för vidare läsning om Mahalanobis distance.

Ett annat sätt att korrigera för korrelationen är att utföra en principalkomponentanalys. Detta kan även reducera antalet dimensioner vilket då också minskar beräkningsbelastningen. Träningsmängden består endast av 398 observationer och stor vikt ligger i modellens sensitivitet. Det anses därmed inte befogat att förenkla modellen och reducera den relativt korta beräkningstiden när detta kan leda till en marginellt lägre sensitivitet.

Det 30-dimensionella hyperplanet går inte att visualisera. Detta försvårar möjligheten att kunna undersöka underliggande orsaker till att den ”falska negativa” observationen klassificeras felaktigt. Figur 8.2 och Figur 8.3 tyder på att den elakartade tumören liknar en godartad tumör med avseende på många av de cellmått som analyserats. Problemet anses därför inte bero på enskilda felaktiga värden till följd av exempelvis felinmatning. Det kan inte motiveras att en förändring av

modellen är önskvärd för att åtgärda detta. För att korrekt klassificera denna observation krävs en betydande lägre precision.

Vidare kan datamaterialet som används i analysen anses bestå av två tydliga klasser och därmed passande för den här typen av analys. Det hade varit av intresse att undersöka hur SVM presterar i andra situationer. Större datamaterial eller datamaterial vars klasser är svårare att separera linjärt kan ge ytterligare kunskap i hur modellering och val av lämplig modell kan angripas. Förslag till fortsatta studier inom ämnet är att undersöka och jämföra hur andra metoder, dels inom maskininlärning och dels inom statistikområdet, presterar. Även om SVM presterar väl på datamaterialet Wisconsin Diagnostic Breast Cancer kan andra metoder fortfarande vara av intresse för att undersöka hur SVM förhåller sig till dessa sett till prestation och komplexitet hos modellen.

Referenser

Abe, S. (2010). *Support Vector Machines for Pattern Classification* (2 ed.). London: Springer London.

Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H.-T. (2012a). *Learning From Data - A Short Course.* AMLBook.com.

Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H.-T. (2012b). e-Chapter 8 Support Vector Machines. In Y. S. Abu-Mostafa, M. Magdon-Ismail, & H.-T. Lin, *Learning From Data.* California Institute of Technology.

Bell, J. (2015). *Machine Learning: Hands-On for Developers and Technical Professionals.* Indianapolis, Indiana, United States of America: John Wiley & Sons, Inc.

Bhavsar, H., & Ganatra, A. (2015). Support Vector Machine Classification using Mahalanobis Distance Function. *International Journal of Scientific & Engineering Research, 6*(1), 618.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* New York, NY: Springer.

Breiman, L. (2001). Statistical Modeling: The Two Cultures. *Statistical Science, 16*(3), 199-231.

Burges, C. J. (1998, Juni). A Tutorial on Support Vector Machines for Pattern. *Data Mining and Knowledge Discovery, 2*(2), 121-167.

Cortes, C., & Vapnik, V. (1995). Support-vector Networks. *Machine Learning, 20*(3), 273-297.

Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines: And Other Kernel-Bases Learning Methods.* Cambridge: Cambridge University Press.

Gunn, S. R. (1998). *Support Vector Machines for Classification and Regression.* University of Southampton.

Guyon, I. (u.å.). *A scaling law for the validation-set training-set size ratio.* Berkeley, California: AT&T Bell Laboratories.

Haasdonk, B., & Pekalska, E. (2009). *Classification with Kernel Mahalanobis Distance Classifiers.*

Hagan, M., Demuth, H., Beale, M., & De Jesús, O. (2014). Generalization. In M. T. Hagan, H. B. Demuth, M. H. Beale, & O. De Jesús, *Neural Network Design* (2 ed.). Martin Hagan.

Hamel, L. H. (2009). *Knowledge Discovery with Support Vector Machines.* Hoboken, New Jersey, United States of America: JohnWiley & Sons, Inc.

Hastie, T., Rosset, S., Tibshirani, R., & Zhu, J. (2004). The Entire Regularization Path for the Support Vector Machine. *Journal of Machine Learning Research, 5*, 1391–1415.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning - Data Mining, Inference, and Prediction* (2 ed.). New York: Springer.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). *A Practical Guide to Support Vector Classification.* Taipei: National Taiwan University.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introducton to Statistical Learning - with Applications in R.* New York, NY: Springer.

Kecman, V. (2001). *Learning and Soft Computing - Support Vector Machines, Neural Networks, and Fuzzy Logic Models.* Cambridge, Massachusetts: MIT.

Ledolter, J. (2013). *Data Mining and Business Analytics with R.* Hoboken, New Jersey: Wiley.

McCarthy, J., & Feigenbaum, E. A. (1990). In Memoriam: Arthur Samuel: Pioneer in Machine Learning. *AI Magazine, 11*(3).

Nationalencyklopedin. (2017). *Klassifikation*. Retrieved 05 10, 2017, from Nationalencyklopedin: http://www.ne.se.ludwig.lub.lu.se/uppslagsverk/encyklopedi/lång/klassifikation

Powers, D. (2011). Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies, 2*(1), 37-63.

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Reasearch and Development, 3*(3), 210-229.

Smith, B. (2004). *Lagrange Multipliers Tutorial in the Context of Support Vector Machines.* Newfoundland: Memorial University of Newfoundland.

Steinwart, I., & Christmann, A. (2008). *Support Vector Machines.* New York, NY: Springer Science+Business Media, LLC.

Wang, L. (2005). *Support Vector Machines: Theory and Applications.* Berlin: Springer.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* New York: Springer.

Bilaga 1 - Vektorer

Vektorer och dess geometriska egenskaper spelar en viktig roll inom SVM. Först introduceras den tvådimensionella vektorn $\mathbf{x} = (a, b)$ som kommer användas som exempel.

En vektors längd definieras med hjälp av Pythagoras sats som

$$\|\mathbf{x}\| = \sqrt{a^2 + b^2}.$$

En vektors riktning definieras som

$$\mathbf{u}_x = \frac{\mathbf{x}}{\|\mathbf{x}\|} = \left(\frac{a}{\|\mathbf{x}\|}, \frac{b}{\|\mathbf{x}\|}\right)$$

eller

$$\mathbf{u}_x = (\cos\theta\,, \sin\theta)$$

där θ är vinkeln mellan vektorn och den horisontella x-axeln.

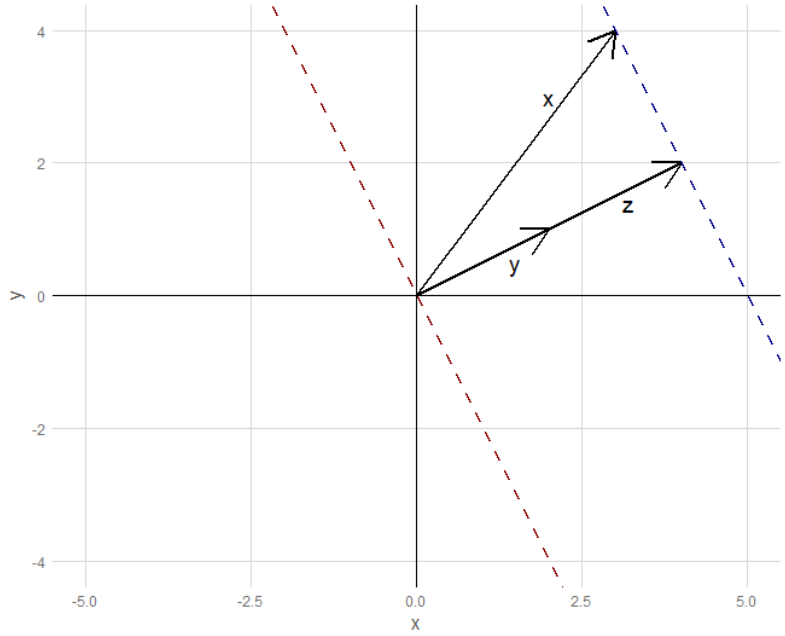
Två nya vektorer införs, **y** och **z**, för att definiera den ortogonala projektionen av en vektor på en annan. I det här fallet är vektorn **z** den ortogonala projektionen av **x** på **y**. Längden av **z** kan med ovan definitioner visas vara

$$\|\mathbf{z}\| = \mathbf{u}_y \cdot \mathbf{x}$$

där $\mathbf{u}_y$ är riktningen av **y**, och därmed också riktningen av **z**. Genom att nyttja $\mathbf{u}_y = \mathbf{u}_z = \mathbf{z}/\|\mathbf{z}\| = \mathbf{y}/\|\mathbf{y}\|$ går det även visa att vektorn **z** ges av

$$\mathbf{z} = \|\mathbf{z}\|\mathbf{u}_z = (\mathbf{u}_z \cdot \mathbf{x})\mathbf{u}_z.$$

Ett exempel relevant till appliceringen av detta inom SVM visualiseras i Figur B1. Här visas hur den ortogonala projektionen **z** av **x** på **y** kan användas för att beräkna avståndet mellan två parallella hyperplan.



Figur B1 *Illustration av ortogonal projektion*

Bilaga 2 - Exempel på kärnfunktion

Detta exempel utgår från en polynomisk kärnfunktion. Den definieras som

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^d. \quad \text{(B1)}$$

Vidare antas att $\mathbf{x} = (x_1, x_2)$ och parametern $d = 2$, vilket ger

$$K(\mathbf{x_i}, \mathbf{x_j}) = (x_{i1}x_{j1} + x_{i2}x_{j2} + 1)^2.$$

Detta uttryck kan utvecklas till

$$1 + x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + 2x_{i1}x_{j1}x_{i2}x_{j2}$$

och sedan skrivas om som en inre produkt av

$$(1, x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}, \sqrt{2}x_{i1}x_{i2})$$

och

$$(1, x_{j1}^2, x_{j2}^2, \sqrt{2}x_{j1}, \sqrt{2}x_{j2}, \sqrt{2}x_{j1}x_{j2}).$$

Redan i detta enkla fall så har dimensionen av vektorn utökats från 2 till 6, utan att summan är beräkningsmässigt tyngre, då det enkelt går att beräknas direkt från uttrycket i Ekvation B1 (Hastie et al., 2009, s. 424).

Bilaga 3 - Kod

```
# Installation och inläsning av paket #################

install.packages(c("tidyverse","e1071","GGally",
                   "ggcorrplot"))
library(tidyverse)
library(e1071)
library(GGally)
library(ggcorrplot)


# Importera och städa dataset ##########################

cancer <- read.csv(url("https://archive.ics.uci.edu/
ml/machine-learning-databases/breast-cancer-wisconsin
/wdbc.data"), header=F)
str(cancer)

# Kontrollera för NAs
sum(is.na(cancer)) # 0

# Ta bort ID-variabel och byt namn på klassvariabeln
cancer <- cancer[,-1]
colnames(cancer) <- c("class", colnames(cancer[2:31]))

# Sampla träningsmängd
set.seed(111)
train <- sample(nrow(cancer),nrow(cancer)*0.7)


# Undersökning av variablerna i datasetet #############

# Scattermatris av 4 utvalda variabler
ggpairs(cancer, columns=c(2,8,12,21),
        aes(color=class, alpha= 0.4), upper=NULL)

# Fördelning av klasstillhörighet
class.summary <- cancer %>%
  dplyr::select(class) %>%
  group_by(class) %>%
  summarise(count = n()) %>%
```

```
  mutate(percent = 100*count/sum(count)) %>%
  arrange(class, desc(class))
class.summary$label =
  paste0(sprintf("%.2f", class.summary$percent), "%")

ggplot(class.summary, aes(x = factor(class), y = count,
                          fill = class)) +
  geom_bar(stat = "identity", width = .7,
           position = position_stack()) +
  geom_text(aes(label = label), size = 4,
            position = position_stack(vjust=0.7)) +
  scale_fill_manual(values=c("#F8766D","#00BFC4"),
                    labels=c("Godartad","Elakartad")) +
  guides(fill=guide_legend(title="Klass")) +
  ylab("Antal") + xlab("Klass")

# Korrelationsmatris
correlations <- cor(cancer[,2:31],method="pearson")
ggcorrplot(correlations, hc.order=T,
           outline.col="white") + xlab("") + ylab("") +
  theme(axis.text.x=element_text(angle=90)) +
  guides(fill=guide_legend(title="Korrelation"))


# Analys: Linjära modeller ############################

# Anpassning av modell (grid search), manuell "leave
# one out"-CV för att kunna kolla korsvaliderings-
# sensitivten

i=1
grid.cost1 <- 10^c(rep(-4:3,1)) # Olika värden på c
lin.mod <- vector("list",length(grid.cost1))
lin.mod.cross <- vector("list",length(train))
lin.mod.cross.pred <- vector("numeric",length(train))
lin.mod.confmat.cross <- vector("list",
                                length(grid.cost1))
for (j in grid.cost1){
  for (k in 1:length(train)){
    lin.mod.cross[[k]] <- svm(class~.,
                              data=cancer[train,][-k,],
                              kernel="linear",cost=j)
    lin.mod.cross.pred[[k]] <-
```

```
      predict(lin.mod.cross[[k]],cancer[train,][k,])
    print(c(j,k))
  }
  lin.mod.confmat.cross[[i]] <-
    table(lin.mod.cross.pred,cancer[train,1])
  lin.mod[[i]] <- svm(class~., data=cancer,
                      kernel="linear",subset=train,
                      cost=j, cross=length(train))
  i <- i+1
}

lin.mod

# Samlar mått från samtliga linjära modeller
# sv: antalet stödvektorer
# cve: korsvalideringsfel
# sens: sensitivitet i korsvalideringen
# cost: värde på c
lin.mod.sv <- vector("numeric",length(lin.mod))
lin.mod.cve <- vector("numeric",length(lin.mod))
lin.mod.sens <- vector("numeric",length(lin.mod))
lin.mod.cost <- vector("numeric",length(lin.mod))
for (k in 1:length(lin.mod)){
  lin.mod.sv[k] <- lin.mod[[k]]$tot.nSV
  lin.mod.cve[k] <- 100-lin.mod[[k]]$tot.accuracy
  lin.mod.cost[k] <- lin.mod[[k]]$cost
  lin.mod.sens[k] <-
    ifelse(nrow(lin.mod.confmat.cross[[k]])==2,
               (lin.mod.confmat.cross[[k]][2,2])/
               sum(lin.mod.confmat.cross[[k]][,2]),0)
}

lin.mod.summary <- data.frame(lin.mod.cve,lin.mod.sv,
                              lin.mod.sens,lin.mod.cost
                              )
colnames(lin.mod.summary) <- c("CVerror", "SV",
                               "sens", "c")
lin.mod.summary$c <- as.factor(lin.mod.summary$c)
lin.mod.summary$x <- as.factor(rep(1,length(lin.mod)))
lin.mod.summary$labelcv <-
  paste0(sprintf("%.1f", lin.mod.summary$CVerror),
         " %")
lin.mod.summary$labelsens <-
```

```
  paste0(sprintf("%.1f", lin.mod.summary$sens*100),
         " %")

# Tileplot över sensitivitet för olika värden på c
linmod.tileplot <- ggplot(lin.mod.summary,
                          aes(y=x,x=c,fill=sens)) +
  geom_tile() + geom_text(aes(label=labelsens)) +
  scale_fill_gradient(low="#f1fff1", high="#66ff99") +
  coord_fixed(ratio = 1, xlim = NULL, ylim = NULL,
              expand = TRUE) + ylab("") +
  theme(axis.text.y=element_blank()) +
  theme(legend.position = "none")

linmod.tileplot


# Analys: Gaussian kernel-modeller ####################

# Anpassning av modell (grid search), manuell "leave
# one out"-CV för att kunna kolla korsvaliderings-
# sensitivten
grid.cost <- 10^c(-2:4)
grid.gamma <- 10^c(-5:0)
rbfdot.mod <- vector("list",length(grid.gamma)*
                       length(grid.cost))
r <- 1
rbfdot.mod.cross <- vector("list",length(train))
rbfdot.mod.cross.pred <- vector("numeric",length(train)
                                )
rbfdot.mod.confmat.cross <- vector("list",
                                   length(grid.gamma)*
                                     length(grid.cost))
for (k in grid.cost){
  for (j in grid.gamma){
    for(i in 1:length(train)){
      rbfdot.mod.cross[[i]] <-
        svm(class~., data=cancer[train,][-i,],
            kernel="radial",cost=k, gamma=j)
      rbfdot.mod.cross.pred[[i]] <-
        predict(rbfdot.mod.cross[[i]],
                cancer[train,][i,])
      print(c(k,j,i))
    }
```

```
    rbfdot.mod.confmat.cross[[r]] <-
      table(rbfdot.mod.cross.pred,cancer[train,1])
    rbfdot.mod[[r]] <-
      svm(class~., data=cancer, kernel="radial",
          subset=train, cost=k, gamma=j,
          cross=length(train))
    r <- r+1
  }
}

rbfdot.mod.confmat.cross
# Samlar mått från samtliga Gaussiska modeller
# sv: antalet stödvektorer
# cve: korsvalideringsfel
# sens: sensitivitet i korsvalideringen
# cost: värde på c
# gamma: värde på gamma
rbfdot.mod.sv <- vector("numeric",length(rbfdot.mod))
rbfdot.mod.cve <- vector("numeric",length(rbfdot.mod))
rbfdot.mod.sens <- vector("numeric",length(rbfdot.mod))
rbfdot.mod.cost <- vector("numeric",length(rbfdot.mod))
rbfdot.mod.gamma <- vector("numeric",length(rbfdot.mod)
                           )
for (k in 1:length(rbfdot.mod)){
  rbfdot.mod.sv[k] <- rbfdot.mod[[k]]$tot.nSV
  rbfdot.mod.cve[k] <- 100-rbfdot.mod[[k]]$tot.accuracy
  rbfdot.mod.sens[k] <-
    ifelse(nrow(rbfdot.mod.confmat.cross[[k]]) == 2,
          ((rbfdot.mod.confmat.cross[[k]][2,2])/
             sum(rbfdot.mod.confmat.cross[[k]][,2])),0)
  rbfdot.mod.cost[k] <- rbfdot.mod[[k]]$cost
  rbfdot.mod.gamma[k] <- rbfdot.mod[[k]]$gamma
}

rbfdot.mod.gamma <- as.factor(rbfdot.mod.gamma)
rbfdot.mod.cost <- as.factor(rbfdot.mod.cost)
rbfdot.mod.summary <-
  data.frame(rbfdot.mod.cve,rbfdot.mod.sv,
             rbfdot.mod.sens,rbfdot.mod.cost,
             rbfdot.mod.gamma)
colnames(rbfdot.mod.summary) <- c("CVerror", "SV",
                                  "sens", "c","gamma")
rbfdot.mod.summary$labelcv <-
```

```
  paste0(sprintf("%.1f", rbfdot.mod.summary$CVerror),
         " %")
rbfdot.mod.summary$labelsens <-
  paste0(sprintf("%.1f", rbfdot.mod.summary$sens*100),
         " %")

# Tileplot för sensitivitet
rbfdot.tileplot <- ggplot(rbfdot.mod.summary,
                          aes(y=gamma,x=c,fill=sens)) +
  geom_tile() + geom_text(aes(label=labelsens)) +
  ylab("gamma") + labs(y=parse(text="gamma")) +
  scale_fill_continuous(name="Sensitivitet",
                        low="#f1fff1", high="#66ff99")

rbfdot.tileplot


# Analys: Val av modeller ##############################

# Bästa modellen väljs ut i varje modellgrupp baserat
# på sensitivitet

# Linjära modeller
best.mod.lin <-
  which(lin.mod.sens %in% sort(lin.mod.sens,
                               decreasing=T)[1])[1]
best.mod.lin.summary <- lin.mod.summary[best.mod.lin,]
best.mod.lin.summary

# Gaussiska modeller
best.mod.rbfdot <-
  which(rbfdot.mod.sens %in% sort(rbfdot.mod.sens,
                                  decreasing=T)[1])[1]
best.mod.rbf.summary <-
  rbfdot.mod.summary[best.mod.rbfdot,]
best.mod.rbf.summary

# Spara de bästa modellerna
best.models <- c(lin.mod[best.mod.lin],
                 rbfdot.mod[best.mod.rbfdot])
```

```
# Analys: prediktering av testset ####################
# och uträkning av prestandamått #####################

# prediktering av testset
ypred <- lapply(best.models,predict,
                cancer[-train,2:31])
confmat <- lapply(ypred,table,cancer[-train,1])

# Sensitivitet
sens=vector("numeric",2)
for(k in 1:length(confmat)){
  sens[k] <- confmat[[k]][2,2]/(sum(confmat[[k]][,2]))
}

sens

# Precision
prec=vector("numeric",2)
k=1
for(k in 1:length(confmat)){
  prec[k] <- confmat[[k]][2,2]/(sum(confmat[[k]][2,]))
}

prec

# Accuracy
accuracy=vector("numeric",2)
k=1
for(k in 1:length(confmat)){
  accuracy[k] <-
    (confmat[[k]][2,2]+confmat[[k]][1,1])/
    (sum(confmat[[k]]))
}

accuracy
1-accuracy # Misclassification rate

# Training error
# Prediktering av träningsset för att hämta E_IN
ypred.train <- lapply(best.models,predict,
                      cancer[train,2:31])
confmat.train <- lapply(ypred.train,table,
                        cancer[train,1])
```

```
accuracy.training=vector("numeric",2)
k=1
for(k in 1:length(confmat)){
  accuracy.training[k] <-
    (confmat.train[[k]][2,2]+confmat.train[[k]][1,1])/
    (sum(confmat.train[[k]]))
}

1-accuracy.training # Training error

# Sammanställning av mått
best.models.scores <-
  data.frame(model=c("linjär","rbf kernel"),
             training.error=(1-accuracy.training),
             cv.error=c(0.01*lin.mod.cve[best.mod.lin],
                        0.01*rbfdot.mod.cve
                        [best.mod.rbfdot]),
             cv.sens=c(lin.mod.sens[best.mod.lin],
                       rbfdot.mod.sens[best.mod.rbfdot]
                       ),accuracy, sens, prec)
best.models
best.models.scores
confmat

# Analys: Undersökning av false negative ##############

# Undersök vilka observationer i testsettet som är
# felklassificerade (med bästa modellen)
which(!(as.numeric(unlist(ypred[2]))==
          as.numeric(cancer[-train,1])))

# En false negative och en false positive,
# kolla vilka av dessa som är false negative
# (faktisk elakartad)
cancer[-train,][c(85,166),] # obs 256

# Funktion för boxplot med false negative markerad
misclassed.obs.boxplot <- function(y){
  ggplot(cancer, aes_string(y=y,x="class")) +
    geom_boxplot() +
    geom_point(data=cancer[256,],
               aes_string(y=y,x="class"),col="red",
               size=3) +
```

```
    theme(legend.position="none") +
    scale_x_discrete(labels=c("G","E"), name=NULL)
}

# Loopa funktionen över samtliga variabler
# (förutom klassvariabeln)
plotcount=1
xindex<-colnames(cancer)[-1]
misclassed.boxplots <- vector("list",
                              length=ncol(cancer)-1)
for(i in xindex){
  misclassed.boxplots[[plotcount]] <-
    misclassed.obs.boxplot(i)
  plotcount=plotcount+1
}

# Funktion som skapar punktdiagram över två variabler
# med den felklassade observationen markerad
misclassed.obs.plot <- function(x,y) {
  ggplot(cancer, aes_string(x=x,y=y,col="class")) +
    geom_point(size=1) +
    geom_point(data=cancer[256,], aes_string(x=x,y=y),
               size=1.5, col="black") +
    theme(legend.position="none")
}

# Loopa över samtliga variabelkombinationer och spara
# till lista
misclassed.plots <-
  vector("list",(length(cancer)-1)*(length(cancer)-2)/2
         )
yindex<-xindex
current=1
length(xindex)
plotcount=0
for(i in xindex[1:29]){
  current=current+1;
  for(j in yindex[current:30]){
    plotcount=plotcount+1
    misclassed.plots[[plotcount]] <-
      misclassed.obs.plot(i,j)
  }
}
```