Maskininlärning och tillämpningar inom genexpressionsanalys

Samuel Rönnqvist
Kandidatavhandling i Datavetenskap
Handledare: Ion Petre och Andrzej Mizera
Tekniska fakulteten
Åbo Akademi
Våren 2010

Referat

Maskininlärning handlar om att med hjälp av algoritmer utvinna kunskap ur en mängd data. Man försöker hitta beskrivande mönster och strukturer, som representeras i form av en modell och kan användas för att åskådliggöra eller fatta beslut utgående från datan.

Datan som ligger i fokus här är numerisk mätdata, som består av mätningar i ett visst antal variabler eller dimensioner. Som ett första steg i analysen presenteras metoder för att visualisera eller förenkla den ofta mångdimensionella datan, utan att förlora viktig information.

I en datamängd där man inte känner till instansernas klasstillhörigheter kan man leta efter strukturer genom att gruppera instanserna i kluster enligt innebördes likhet. I fall klasstillhörigheterna däremot redan är bekanta, kan datan användas för att träna modeller som kan klassificera obekant data. Dessa paradigmer kallas oövervakad respektive övervakad inlärning och utgör det huvudsakliga fokuset i denna översikt.

Exempel på tillämpningar av olika maskininlärningsmetoder ges inom molekylärbiologins domäner. I och med den snabba utvecklingen av bioteknologisk mätutrustning, som till exempel Microarray-chips, har mängden data ökat explosionsartat de senaste åren. Närmare presenteras hur mätningar av geners expressionsnivåer (tillfällig aktivitet) i vävnad kan analyseras med hjälp av maskininlärning, för att bistå forskning och diagnostisering av sjukdomar.

Nykelord: maskininlärning, mönsterigenkänning, dataanalys, genexpression, Microarray

Innehåll

1. Inledning

Maskininlärning (eng. machine learning) är den disciplin inom datavetenskapen som undersöker hur man med hjälp av algoritmer kan få datorer att känna igen mönster och lära sig från data. Målet är att åstadkomma modeller som kan användas av en mänska för att bättre förstå en ofta stor och komplex datamängd, genom att visualisera eller utvinna relevant information ur den. I vissa fall är man intresserad av att konstruera algoritmer som kan klassificera objekt eller förutspå numeriska värden. Allt detta åstadkoms genom att söka efter naturliga strukturer och lära upp generaliserade modeller ur en mängd data.

Mänskan och naturen står ofta som modeller när maskininlärningsalgoritmer utvecklas, exempelvis försöker artificiella neurala nätverk på ett förenklat vis imitera hjärnan. I många fall är mänskan överlägsen alla algoritmer vad gäller snabbhet, flexibilitet och precision i att kunna skilja ur detaljer i sin omgivning. Algoritmernas fördelar är däremot automationen och förmågan att kunna ta väldigt stora datamängder i beaktande.

Det finns naturligtvis många områden inom vilka maskininlärning kan tillämpas. Som en följd av den snabba utvecklingen av bioteknologisk mätutrustning under de senaste åren finns det nu gigantiska mängder biologisk data att tillgå. Detta motiverar användningen av maskininlärning, för att kunna analysera denna oftast väldigt komplexa och omfattande data. Analys av genexpression är ett viktigt delområde, som kan bidra till att öka förståelsen för hur livet fungerar. (Tarca, s. 1; Duda, s.1-9)

1.1 Syfte och fokus

Denna avhandling är en översikt över de centralaste paradigmerna inom maskininlärning. Inom varje paradigm presenteras några representativa metoder för att åskådliggöra olika principer. Det är meningen att läsaren, med en dataveten-skaplig bakgrund, skall få en uppfattning om några av maskininlärningens mer grundläggande idéer samt hur dessa kan tillämpas praktiskt.

Fokuset ligger på analys av numerisk data, vilket begränsar urvalet av metoder. Det finns syntaktiska maskininlärningsmetoder som lämpar sig för analys av symbolsekvenser, t.ex. dolda Markov-modeller (eng. Hidden Markov Models) är

populära för röstigenkänning (Valafar, s. 9), men dessa behandlas inte här. Exempel på tillämpningar ges inom analys av data från bioteknologisk mätutrustning, närmare bestämt analys av genexpressionsdata från Microarray-chips.

Baldi & Brunak anser att det finns åtminstone tre olika nivåer att analysera genexpressionsdata på. Den enklaste nivån av analys är att undersöka enskilda geners skillnader i expressionsnivå en i taget, mellan kontroll- och behandlingssituationer. På den andra nivån undersöker man istället flera gener för att t.ex. hitta mönster av samexpression. Den tredje nivåns analys försöker hitta nätverk mellan gener som beskriver de observerade expressionsmönstren. (Baldi & Brunak, s. 300)

Metoderna som presenteras här håller sig främst på den andra nivån och i mindre utsträckning på den första. För analys på den tredje nivån kan man använda sej av Bayesianska nätverk (Valafar, s. 5; Ott), hela denna nivå ligger dock utanför fokuset för den här översikten.

1.2 Avhandlingens struktur

Denna avhandling inledes med presentation av de mindre komplicerade tillvägagångssätten för dataanalys för att successivt gå mot mer komplicerade. Kapitel 2 handlar om reduktion av datans dimensionalitet, metoder för att göra den mera förståelig eller filtrera bort irrelevanta detaljer. Sedan presenteras i kapitel 3 metoder för att försöka hitta naturliga strukturer i okänd data. I kapitel 4 antas däremot att bekant data finns att tillgå, vilket kan utnyttjas för att lära upp en modell som kan känna igen liknande mönster i okänd data och klassificera den.

Alla dessa kapitel beskriver grundläggande principer och metoder inom varje paradigm ur ett generellt datateoretiskt perspektiv. Men de avslutas med exempel på hur denna teori kan tillämpas för analys av genexpressionsdata. Sektion 1.3 ger en del grundläggande biologisk bakgrundskunskap som är nödvändig för att förstå tillämpningarna, samt en kort beskrivning av Microarray-chips.

Avslutningsvis tar kapitel 5 upp hur den kunskap som erhålls med hjälp av maskininlärning kan utvärderas såväl ur ett datateoretiskt som domänspecifikt perspektiv.

1.3 Biologisk bakgrund

För att förstå tillämpningarna som presenteras i denna avhandling krävs en grundläggande förståelse av vissa molekylärbiologiska begrepp och tekniken som berörs. Nedan ges en kort förklaring av dessa.

1.3.1 Gener och proteiner

En gen är en del av en organisms DNA-sekvens som beskriver hur en biologisk produkt, t.ex. ett protein, skall syntetiseras (Nelson & Cox, s. 271). Ett protein är en lång kedja av aminosyror, som kan ha en funktionell eller strukturell roll i en organism. Syntesen sker enligt informationen som finns kodad i genen och som överförs med hjälp av RNA. Produkter av proteinsyntesen är bl.a. enzymer, hormoner och anti-kroppar (Valafar, s. 2; Nelson & Cox, s. 283).

1.3.2 Genexpression

Att en gen expresseras betyder att den avskrivs till RNA (Nelson & Cox, s. 315). Varje cell i en organism innehåller den fullständiga DNA-sekvensen, genomet. Dock är endast en bråkdel av alla gener expresserade i en cell på en gång. Var, när och i vilken grad en gen expresseras bestäms av en mekanism kallad differentierad genexpression. Denna regleringsmekanism ger även upphov till olika celltyper utgående från samma genom (Valfar s. 2-3; Nelson & Cox s. 1115). I denna avhandling syftar begreppet genexpression i allmänhet på differentierad genexpression.

1.3.3 Microarrays

DNA-Microarray-chips är bioteknologiska mätinstrument som kan användas för att mäta expressionsnivåer för tusentals gener samtidigt. En Microarray består av en yta till vilken DNA-molekyler är bundna. Från ett vävnadsprov utvinner man nukleinsyresekvenser som sedan märks, så att de kan fluoresceras med hjälp av laser. För att mäta genexpressionen använder man nukleinsyresekvenser från mRNA. Nukleinsyrorna binder till DNA-fragmenten på chipsytan. Specifika nukleinsyresekvensers förekomst kan sedan avläsas fotografiskt tack vare deras fluorescens. (Stekel, s. xi, 1, 10, 14)

Efter förprocessering av Microarray-avläsningarna fås en datatabell som är redo för analys. Denna tabells kolumner består av alla de gener som avlästs och raderna av olika prover, tagna under olika förhållanden (Tarca, s.2).

2. Reduktion av dimensionalitet

Varje instans i en mängd mätdata består av ett antal mätvariabler eller attribut (eng. feature). Dessa attribut utgör alla varsin dimension. Datans dimensionalitet kan ofta bli väldigt hög, speciellt vad gäller Microarray-data som typiskt innehåller 10'000-tals geners olika expressionsnivåer (Baldi & Hatfield, s. 73; Castells, s. 4).

Ett sätt att minska antalet dimensioner är sk. attributextraktion (eng. feature extraction), vilket används inom t.ex. ansiktsigenkänning för att hitta ett ansikte i en bild. Extraktionen försöker eliminera stora men irrelevanta variationer mellan olika datainstanser, för att på så vis förbättra prestandan för den efterföljande maskininlärningen. Genexpressionsdatan som ligger i fokus här är ursprungligen extraherad ur en bild av Microarray-chipset (se 1.3.3), men här antas datan bestå av en färdigt förprocesserad och normaliserad tabell av genexpressionsnivåer per gen och prov.

Reduktion av dimensionaliteten kan även användas för att visualisera mång-dimensionell data eller för att minska risken för överanpassning (eng. overfitting). Överanpassning betyder att den tränade modellen inte är tillräckligt generell, att den är överanpassad till en viss mängd träningsdata och fungerar dåligt för okänd data.

Oftast är bara en del av alla attribut informativa i ett visst avseende, medan andra är obetingade och kan betraktas som brus. Det finns två huvudsakliga tillvägagångssätt för dimensionalitetsreduktion, attributval och attributomvandling. (Tarca, s. 5; Larrañaga, s. 4; Stekel, s. 151-152; Duda, s. 11-12, 16, 72)

2.1 Attributval

Attributval (eng. feature selection) går ut på att försöka plocka ut de attribut som är mest informativa i avseende på den analys man företar sig. Nedan förklaras två grupper av metoder, filter- och inpackningsmetoder.

2.1.1 Filtermetoder

Filtermetoder försöker filtrera bort statistiskt irrelevanta attribut. Vid jämförelse mellan olika dataklasser kan man undersöka enskilda attributs variation mellan klasserna. Sedan kan alla attribut vars variation inte överskrider en viss

tröskelfaktor filtreras bort.

En nackdel med denna metod är att samverkan inom en grupp attribut inte tas i beaktande, således hamnar denna metod i Badli & Brunaks första nivå av analys (se sektion 1.1) (Tarca, s. 6; Baldi & Hatfield, s. 54).

2.1.2 Inpackningsmetoder

Inpackningsmetoder (eng. wrapper methods) undersöker enskilda eller flera attribut samtidigt, en övervakad inlärningsmetod guidar attributvalet. Övervakade inlärningsmetoder, som diskuteras i kapitel 4, kan testas automatiskt för att bestämma en felfrekvens som är ett mått på metodens precision. Attributen kan väljas slumpmässigt med hjälp av en genetisk algoritm, där val som resulterar i en förbättrad precision bibehålls.

Denna metod kan upptäcka grupper av samverkande attribut och dessutom optimera attributvalet specifikt för den övervakade inlärningsmetod som används. Däremot är denna metod beräkningsbart tung. (Tarca, s. 6)

2.2 Attributomvandling

Attributomvandling går ut på att skapa en maximalt informativ men lågdimensionell representation av en datamängd, genom att skapa en ny kombinerad mängd attribut utgående från de gamla. Här presenteras en av de vanligaste metoderna, principalkomponentanalys. (Tarca, s. 5).

Principalkomponentanalys (eng. Principal Component Analysis), förkortat PCA, är en metod som skapar en lågdimensionell representation av högdimensionell data. För att projektionen av datarymden, i en lägre dimension, skall vara så informativ som möjligt görs den i en sådan vinkel att variansen (spridningen) blir maximal. Detta kan jämföras med att ta en 2-dimensionell bild av ett 3-dimensionellt objekt ur en sådan vinkel att det ser så stort ut som möjligt på bilden.

PCA söker den första principalkomponenten, projektionen av den N-dimensionella datarymden i den vinkel som ger störst varians i N-1 dimensioner. Den andra komponenten, som är vinkelrät mot den första, söks på samma sätt i den nya N-1 dimensionella rymden. Proceduren upprepas tills det önskade antalet principalkomponenter har hittats.

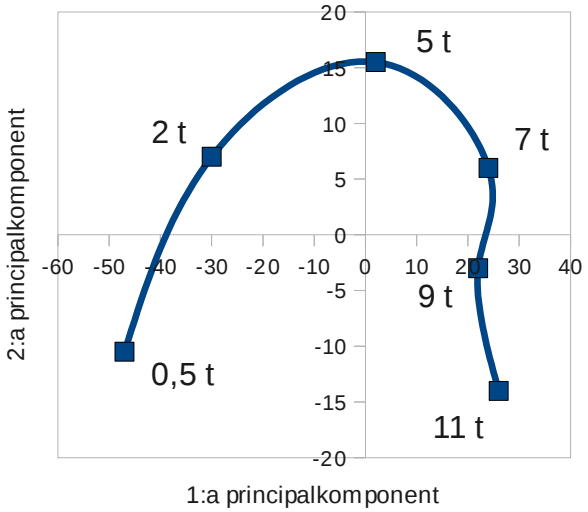
De resulterande komponenterna är de nya axlarna i den nya lågdimensionella representationen av den ursprungliga högdimensionella datarymden. Den första komponenten är den mest informativa, därefter följer resten i fallande ordning.

Genom att rita upp de två eller tre första resulterande komponenterna kan man enkelt visualisera datamängden (se 2.3). PCA kan också användas före andra maskininlärningsmetoder för att förbättra deras prestanda, genom att eliminera mindre signifikanta variationer i datan (se 4.5). (Witten, s. 306-309; Stekel, s. 152-155)

2.3 Tillämpning: Visualisering av tidsseriedata med PCA

Stekel presenterar ett experiment där 6 st genexpressionsmätningar har gjorts på jästceller under deras sporbildningsfas. Expressionsnivåerna av 6'000 gener finns registrerade, datan filtreras så att de 1'000 mest differentierade generna lämnar kvar för att sedan analyseras med hjälp av PCA.

De två första principalkomponenterna används för att rita upp figur 2.1. De olika proverna finns utmärkta med den relativa tiden i timmar då de är tagna. Provernas placering följer ett tydligt mönster som kan antas ha en biologisk betydelse.



Figur 2.1: Genexpressionsprover ur en tidsserie under en jästcells sporbildningsfas, uppritade i de två första principalkomponenterna.

Stekel tolkar mönstren bl.a. som att den lilla variationen i 1:a komponenten mellan 7 och 11 timmar tyder på att sporbildningsfasen då är avslutad. Den andra komponenten visar en förändring fram till 5 timmars provet, följt av en återgång. Detta skulle då tyda på en kortare övergående process som är aktiv endast under

de 5 första timmarna av sporbildningsfasen.

PCA-metoden ger även ett mått på hur stor del av variabiliteten i datan som behålls i de olika principalkomponenterna. I detta experiment bibehålls 77% av variabiliteten i 1:a komponenten och 88% i de två första. Detta menar Stekel är ett bevis på att de biologiska processerna bakom sporbildning är relativt enkla. Redan fem principalkomponenter räcker för att täcka hela datans variabilitet. (Stekel, s. 154-155)

3. Oövervakad inlärning

Oövervakad inlärning (eng. Unsupervised Learning) handlar om att upptäcka naturliga grupperingar i data. Objekten i en datamängd delas in i grupper, eller kluster, enligt deras likhet sinsemellan. Detta sker enligt en given definition av likhet, t.ex. Euklidisk distans som definieras i ekvation 3.1 för *p* dimensioner. (Tarca, s. 2, 6; Larrañaga, s. 9)

$$d(\vec{x},\vec{y})=\sqrt{\sum_{i=1}^{p}(\vec{x}_i-\vec{y}_i)^2} \quad (3.1)$$

Man känner varken till några klasstillhörigheter för objekten eller antalet olika klasser som finns representerade i datan. Processen måste därför vara självguidad, eller oövervakad (Tarca, s. 2; Larrañaga, s. 9). Till de populäraste teknikerna för klusteranalys, som den oövervakade inlärningen ofta kallas, hör partitionella och hierarkiska metoder (Tarca, s. 6; Valfar, s. 5). Båda dessa metoder presenteras nedan.

3.1 Partitionella metoder

Partitionering eller delande klusteranalys (eng. Divisive Clustering) försöker dela upp datarymden, så att alla objekt tilldelas ett kluster. Många metoder kräver att antalet kluster som skall konstrueras fördefinieras. Bland dem k-Means algoritmen som presenteras här och som är en av de mest använda. Det finns också andra algoritmer som automatiskt försöker hitta det lämpligaste antalet kluster, men dessa behandlas inte här. (Larrañaga, s. 9; Valfar, s. 5)

För att initiera k-Means algortimen anger användaren antalet klustercentroider k och utgångspositionerna för dessa sätts slumpmässigt. Sedan tilldelas varje datapunkt den centroid som ligger närmast. För varje kluster beräknas en ny

centroid, eller mittpunkt, enligt dess nya uppsättning medlemmar. Algoritmen itererar dessa två steg, så att alla objekt tilldelas rätt kluster enligt de nya centroiderna. När stabila kluster har uppnåtts terminerar algoritmen, dvs. när inga datapunkter mera byter kluster. (Witten, s.137; Duda, s. 526-527; Larrañaga, s. 9)

Algoritmens mål är att för varje kluster *k* minimera uttrycket i ekvation 3.2, där *d* är en distansfunktion, c_i centroiden och p_i mängden datapunkter i klustret. Resultatet som fås är dock ett lokalt minimum som beror på den slumpmässiga initieringen. Oftast hjälps detta problem med att köra algoritmen flera gånger och välja det bästa resultatet (Witten, s. 137-138). Det kan vara svårt att hitta naturliga optimala kluster med hjälp av k-Means eftersom det lämpliga antalet kluster inte på förhand är känt. (Valfar, s. 6)

$$\sum_{n \in k} d(c_i, p_{i,n})^2 \quad (3.2)$$

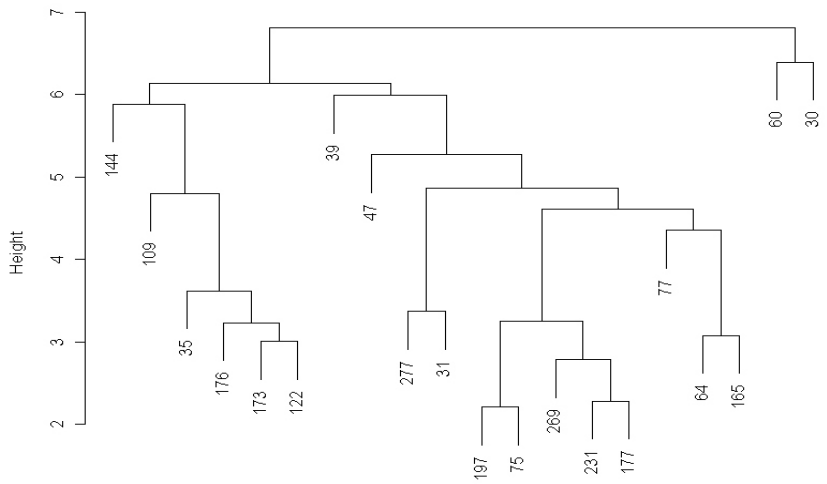
Istället för den trivialare Euklidiska distansen används även ibland Mahalanobis distans som distansfunktion, definierad i ekvation 3.3. Den försöker inte bara minimera de inre distanserna i ett kluster, utan också maximera distansen mellan alla kluster (Valfar, s. 6).

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x}_k - \vec{y}_i)^T \sum_i^{-1} (\vec{x}_k - \vec{y}_i)} \quad (3.3)$$

3.2 Hierarkiska metoder

Ofta är det naturligare att ordna datagrupper i en hierarki än bara platt i en nivå. I hierarkisk klusteranalys kan ett kluster innehålla underkluster, istället för att alla kluster ligger bredvid varandra i ett plan. Den hierarkiska grupperingen kan representeras som ett träddiagram, kallat dendrogram (se figur 3.1).

På den lägsta nivån är varje datapunkt ett kluster, medan den högsta nivån består av endast ett kluster. När man går från en lägre till en högre nivå kommer närliggande kluster successivt att sammanföras i ett gemensamt kluster på nästa nivå uppåt. Distansen, eller likheten, mellan varje kluster i hela trädet är mätbar. Detta kan vara till hjälp för att hitta naturliga uppdelningar av datan.



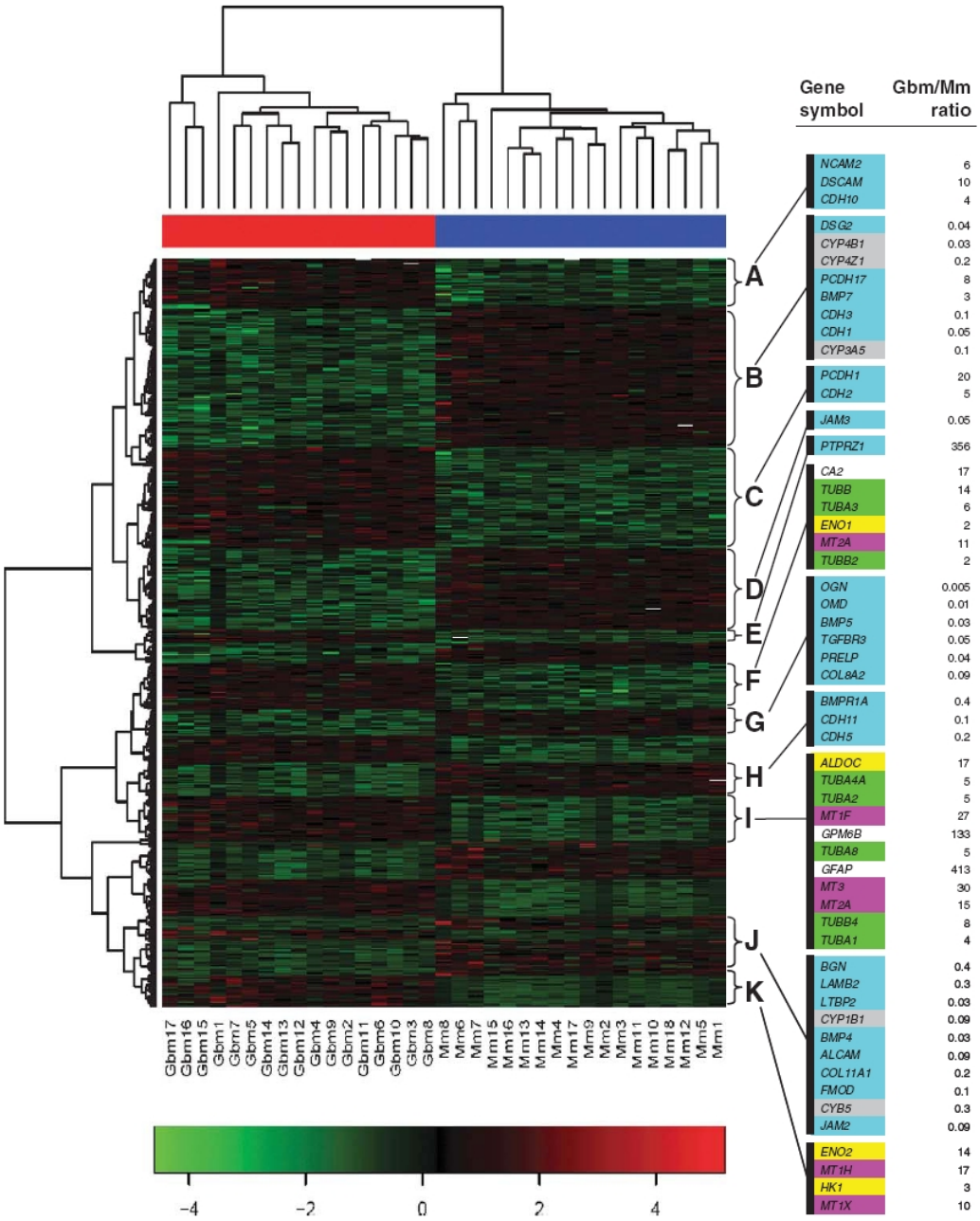
Figur 3.1: En dendrogram-representation av en hierarkisk klusteranalys, siffrorna i löven representerar olika dataobjekt.

Hierarkiska metoder kan delas in i agglomerativa och delande. Agglomerativa metoder bygger dendrogrammet nerifrån upp, börjande med ett kluster per datapunkt och slutande med ett kluster på högsta nivå. Delande metoder bygger däremot uppifrån ner, börjande med ett globalt kluster som successivt delas tills lägsta nivån nås. Agglomerativa metoder är populärare eftersom delande klusteranalys oftast är beräkningsbart ineffektiv. (Duda, s. 550-553; Larrañaga, s. 10)

Det finns olika sätt att mäta avståndet mellan kluster när man bygger hierarkin agglomerativt, några av de vanligare presenteras kort här. Distansen mellan två kluster kan definieras som avståndet mellan deras närmast liggande eller mest avlägsna medlemmar. Dessa två sätt benämns enkellänkande (eng. single-link) respektive fullständigt länkande (eng. complete-link). En annan metod som är snabbare och mindre känslig för mycket avvikande medlemmar (eng. outliers) använder avståndet mellan centroiderna. Wards metod söker det klusterpar vars gemensamma inre distanssumma utgör en så liten ökning som möjligt. (Larrañaga, s. 10; Duda, s. 553-555).

3.3 Tillämpning: Biklusteranalys av genexpressionsdata

De typer av klusteranalys som har presenterats i detta kapitel kan användas för att upptäcka grupper av samexpresserade gener, men de kan inte hitta gener som är samexpresserade under vissa förhållanden och oberoende under andra. Biklusteranalys löser detta genom att hierarkiskt gruppera och ordna både raderna och kolumnerna, dvs. proverna och generna, i datamängden samtidigt. (Tarca, s. 6)



Figur 3.2: En värmekarta som representerar en biklusteranalys av hjärntumörsprover (kolumner) och deras mest differentierade gener (rader). Färgskalan indikerar expressionsnivåerna.

© Castells m.fl., bilden reproducerad med tillstånd.

Olika delmängder av proverna representerar olika förhållanden, dessa kan identifieras genom deras samexpresserade gener. Med ett förhållande kan avses t.ex. en viss sjukdom, eller en specifik undertyp.

Castells m.fl. har använt biklusteranalys på genexpressionsdata från hjärntumörer av typerna Glioblastom och Meningeom, för att hitta gengrupper som kan användas för att skilja på de båda tumörtyperna. Figur 3.2 visar en visualisering

av biklusteranalysen med hjälp av en sk. värmekarta (eng. heat map). De 629 mest differentierade generna är ordnade i rader och de 35 olika tumörproverna i kolumner.

Till höger har genkluster märkts enligt deras funktionella familjer eller biologiska signalräckor som de är relaterade till. Från värmekartan ser man hur expressionsnivåerna, som indikeras av färgskalan, för en viss gengrupp korrelerar med tumörtyp. De differentierande gengrupperna kan studeras vidare på ett biologiskt plan för att försöka förstå mekanismerna bakom de olika cancertyperna. Resultaten från biklusteranalysen kan också användas för att automatiskt klassificera mellan Glioblastom- och Meningeom-tumörer utgående från ett Microarray-prov. I sektion 4.5 ges ett annat exempel på hur tumörer kan klassificeras automatiskt. (Castells, s. 6-7)

4. Övervakad inlärning

Inom oövervakad inlärning (se kaptiel 3) kan klasser i data hittas automatiskt i form av kluster, men dessa är inte garanterat korrekta indelningar och har inte heller beskrivande namn. Inom övervakad inlärning (eng. Supervised Learning) utnyttjar man däremot definierade klasser och kända klasstillhörigheter hos instanserna i en datamängd för att träna upp en klassificerare.

En klassificerare (eng. classifier) är en modell som kan bestämma en datainstans' klass baserat på dess attribut (variabler), efter att den tränats med exempeldata och tillhörande klassetiketter. Till övervakad inlärning räknas även prediktion av numeriska värden med hjälp av tränade modeller, men detta behandlas inte närmare här. (Tarca, s. 1-2; Larrañaga, s. 3)

Först diskuteras här en del grundläggande principer inom klassificering, följt av några representativa metoder. Det är viktigt att komma ihåg att olika inlärningsmetoder lämpar sig olika väl för olika problem. Det finns ingen generellt bästa metod, utan det varierar enligt datans natur. (Larrañaga, s. 5)

4.1 Träning och testning

Givet en datamängd med kända klassetiketter, bör denna delas upp i tränings- och testmängder innan en klassificerare kan börja tränas. Träningsmängden används av den inlärningsalgoritm man använder för att anpassa, eller lära upp, klassifi-

ceraren. Den klassificerar sedan datainstanser ur testmängden och resultatet jämförs med den kända klassetiketten. På så vis kan man bestämma en felfrekvens som ett mått på klassificerarens precision.

Det är viktigt att komma ihåg att denna felfrekvens bara är teoretisk och kan vara för optimistisk vad gäller klassificering av okänd data. Speciellt om samma data som använts för träning också användes för testning, skulle felfrekvensen bli helt otillförlitlig. I vissa fall behövs även en tredje sk. valideringsmängd, t.ex. för att kunna optimera en klassificerare. Ett exempel på detta ingår i tillämpningsexemplet i sektion 4.5. I kapitel 5 diskuteras strategier för uppdelning när datamängden man har att tillgå är väldigt liten. (Larrañaga, s. 4; Tarca, s. 2; Witten, s. 144-146)

4.2 Probabilistisk grund för klassificering

Bayesiansk beslutsteori (eng. Bayesian decision theory) erbjuder en viktig teoretisk grund för klassificering. Här beskrivs några grundläggande sannolikhetsteoretiska begrepp samt Bayes' beslutsregel som beskriver klassificering i probabilistiska termer.

Den obetingade sannolikheten att ett objekt tillhör en viss klass kallas för klassens *a priori* sannolikhet. I det triviala fallet där man inte har någon information att tillgå om de objekt som skall klassificeras, utan endast har klassernas a priori sannolikheter att tillgå, klassificeras alla objekt konsekvent till den mest sannolika klassen. Felfrekvensen blir förstås väldigt hög ifall att den valda klassens sannolikhet inte är mycket större än summan av alla andra klassers sannolikheter.

För att kunna göra en mera tillförlitlig klassifikation behövs någon form av observation av objektet som skall klassificeras. Den observerade informationen om ett objekt kan tänkas vara en numerisk variabel x, som här anses vara stokastisk enligt en känd täthetsfunktion $p(x)$ (även kallad bevis). För klassificering är det nödvändigt att även känna till de klassbetingade sannolikheterna $p(x|c_i)$ för varje klass c_i, dvs. sannolikheten $p(x)$ när man vet att ett specifikt c_i har inträffat.

Vid klassificering söker man den klass c_i som är mest sannolik givet en observation x. Bayes' formel ger denna *a posteriori* sannolikhet $P(c_i|x)$ enligt ekvation 4.1.

$$P(c_i|x) = \frac{p(x|c_i)P(c_i)}{p(x)} \quad (4.1)$$

Målet för klassificeringen kan också definieras som att välja en klass så att sannolikheten för felval minimeras. Sannolikheten för ett felval är summan av a posteriori sannolikheterna för de klasser som inte valdes. Eftersom summan av alla a posteriori sannolikheter är 1, kan felsannolikheten även bestämmas enbart av den valda klassens a posteriori sannolikhet. Ekvation 4.2 definierar den *x*-betingade sannolikheten för felval, för den valda klassen *c'*.

$$P(fel|x) = \sum_{c \in D} P(c|x) = 1 - P(c'|x), D = \{C - c'\} \quad (4.2)$$

Den slutgiltiga klassificeringsregeln kan skrivas som ekvation 4.3, vilken ger den mest sannolika klassen givet en observation *x*. Nämnaren *p(x)* kan här utelämnas ur Bayes' formel eftersom den är samma för alla klasser.

$$g(x) = \underset{c}{argmax}\, p(x|c)P(c) \quad (4.3)$$

Denna metod är en enkel modell för klassificering av objekt i en mängd klasser, som förutsätter att alla sannolikheter är kända. I fall att flera variabler används, antas de alla vara oberoende av varandra och därför kallas den ofta för en naiv Bayes klassificerare. (Duda, s. 20-23; Larrañaga, s. 5)

4.3 Distansbaserade metoder

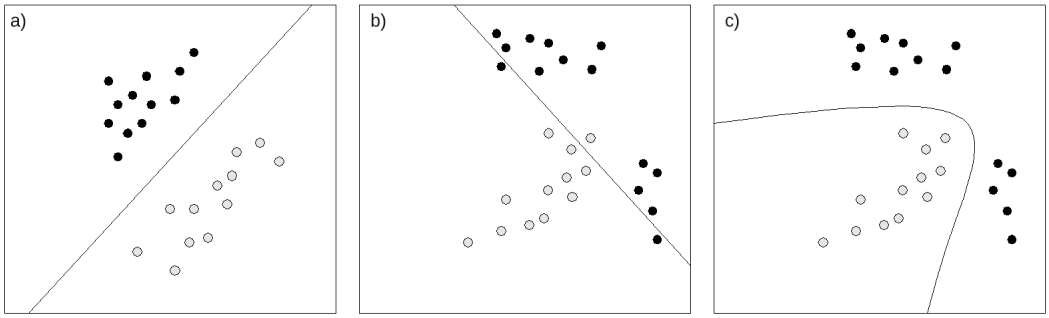
Liksom distanser mellan datapunkter används för att guida oövervakad inlärning kan de även utnyttjas för klassificering. En av de vanligaste distansbaserade metoderna kallas för närmaste granne (eng. nearest neighbour) och förkortas NN. Den försöker inte hitta sannolikheter som i föregående sektion, utan direkt definiera a posteriori sannolikheterna för varje klass.

NN-metoden klassificerar en okänd datapunkt genom att mäta distansen från den till alla punkter i träningsmängden och klassificera den enligt den närmaste punktens klass. För att göra metoden mera robust kan man klassificera enligt de *k* närmaste punkternas majoritetsklass, då kallas metoden k-NN.

Distansfunktionen är typiskt Euklidisk distans, men liksom inom oövervakad inlärning kan andra definitioner på distans användas även här. (Larrañaga, s. 6-7; Duda, s. 161)

4.4 Perceptron-baserade metoder

Om datapunkterna från de olika klasserna kan skiljas åt med hjälp av ett hyperplan säger man att datamängden är linjärt separabel. I detta fall kan man hitta en linjär diskriminantfunktion som kan fungera som klassificerare, perceptron-algoritmen som presenteras här konstruerar en sådan. Figur 4.1 visar exempel på en linjärt separabel datamängd (a) och en som inte är det (b). Efter att perceptronen har beskrivits diskuteras hur perceptroner kan kombineras så att den senare datamängden kan separeras (c).



Figur 4.1: a) Ett linjärt hyperplan i 2 dimensioner som separerar två klasser. b) Två linjärt oseparabla klasser. c) En icke-linjär separation som kan åstadkommas med ett neuralt nätverk

Ett hyperplan i p dimensioner definieras av ekvation 4.4. Datans attributvektor betecknas a och w är dess viktvektor. a_0 är ett extra attributvärde som är konstant 1 och skapar tillsammans med sin vikt w_0 en sk. bias, en justeringskonstant.

$$\vec{w}_0\vec{a}_0+\vec{w}_1\vec{a}_1+...+\vec{w}_p\vec{a}_p=0 \quad (4.4)$$

Ekvation 4.5 definierar en linjär diskriminantfunktion som kan tränas att klassificera linjärt separabel data. I det triviala fallet med två klasser bestäms klasstillhörigheten av på vilken sida om hyperplanet datapunkten hamnar, dvs. om $g_c(x)$ är större eller mindre än 0. Alternativt kan det ses som att klassificeraren anger om en vektor tillhör, eller inte tillhör, den klass den är tränad för.

$$g_c(\vec{x})=\sum_{i=0}^{p}\vec{a}_i\vec{x}_i=\vec{a}^T\vec{x} \quad (4.5)$$

I ett fall med flera klasser kan klassificeraren definieras av ekvation 4.6. Varje klass har en egen linjär diskriminantfunktion $g_c(x)$ och klassen bestäms enligt vilken som ger högst värde. Denna funktion kan jämföras med Bayes-klassifi-

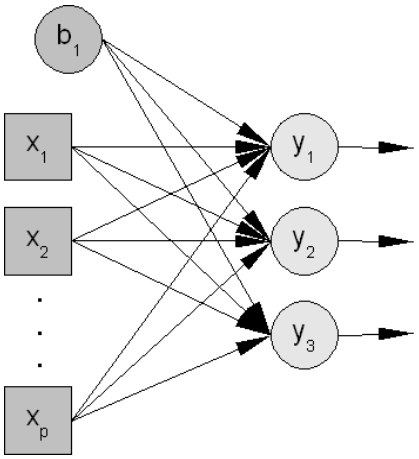
ceraren i ekvation 4.3, här har diskriminantfunktion ersatt a posteriori sannolikheten som baserades på de kända täthetsfunktionerna. (Witten, s. 124-125; Duda, s. 215-219; Tarca, s. 9)

$$g(\vec{x}) = \underset{c}{argmax}\, g_c(\vec{x}) \quad (4.6)$$

4.4.1 Enkellagers-perceptron

Algoritmen för att träna en enkellagers-perceptron (eng. single-layer preceptron) söker ett separerande hyperplan för en datamängd, genom att justera vikterna tills hyperplanet klassificerar träningsdatan korrekt.

Perceptronen har en ingång för varje attribut samt en för biasen, den illustreras ofta som i figur 4.2 där tre perceptroner utgör ett nätverk. Varje ingång är associerad med en vikt, som i ekvation 4.5. Vikternas startvärden är 0.



Figur 4.2: Tre perceptroner i ett enkellagersnätverk som kan klassificera i tre klasser.

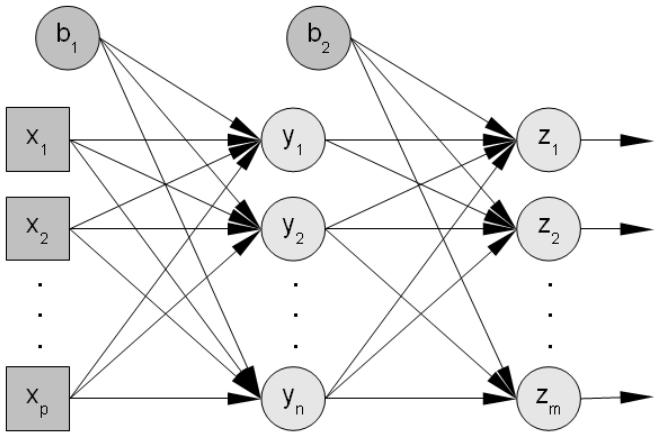
Det finns flera olika variationer på perceptron-algoritmen, en av de mer typiska förklaras här. Träningen är en iterativ process som börjar med att alla vektorer i träningsmängden klassificeras och resultatet jämförs med etiketterna. Mängden X av alla vektorer som klassificerats fel används för att uppdatera viktvektorn enligt ekvation 4.7. En inlärningsgrad η med ett värde mellan 0 och 1, t.ex. 0,5, används för att uppnå en generaliserad viktvektor. Algoritmen terminerar när alla tränings-vektorer klassificeras rätt. (Duda, s. 223-235)

$$\vec{w}' = \vec{w} + \eta \sum_{x \in X} \vec{x} \quad (4.7)$$

4.4.2 Flerlagers-perceptron

Linjär separabilitet mellan klasserna i en datamängd som är kravet för en linjär diskriminantfunktion är ett specialfall som oftast inte gäller för mer komplex data. Genom att kombinera perceptroner i ett flerlagersnätverk kan dock även linjärt oseparabla mängder separeras. Dessa nätverk kallas oftast för artificiella neurala nätverk och perceptronen för neuron eller enhet.

Figur 4.3 visar en schematisk bild av ett typiskt neuralt nätverk, av sk. feed-forward typ där alla noder är kopplade till alla noder i nästa lager. Attributvektorn fungerar som indata till ingångarna i den vänstra kolumnen, bias-noderna *b* har det konstanta värdet 1. Varje pil har också här en associerad vikt. I mitten finns ett eller ibland två sk. dolda lager av neuroner, tack vare detta extra lager kan det neurala nätverket göra icke-linjära separationer. Till höger finns utgångs-neuronerna som har samma funktion som i ett enkellagers perceptronnätverk.



Figur 4.3: Modell för ett artificiellt neuralt nätverk som klassificerar p-dimensionella vektorer i m klasser med n neuroner i ett dolt lager.

Neuronen skiljer sig dock i att utdatan är en monotont ökande kontinuerlig funktion av den tidigare beskrivna perceptronens utvärde. Vanligen är det en icke-linjär sigmoid-funktion (ekvation 4.8), som går brant från 0 till 1 för värden kring 0 i en form som påminner om ett S.

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (4.8)$$

Ett neuralt nätverk av feed-forward-topologi med tre lager, som i figur 4.3, kan matematiskt definieras som ekvation 4.9. Givet en attributvektor *x* definierar

funktionen $g_z(x)$ utvärdet för en neuron z i utdatalagret. Ekvation 4.6 kan användas för att kombinera alla $g_z(x)$ till en klassificerare. Viktvektorn mellan första och andra lagret benämns w_y och mellan andra och tredje w_z.

$$g_z(\vec{x}) = \sigma\left(\sum_{j=1}^{n} \vec{w}_{z,j}\,\sigma\left(\sum_{i=1}^{p} \vec{w}_{y,i}\,\vec{x}_i + \vec{w}_{y,0}\right) + \vec{w}_{z,0}\right) \quad (4.9)$$

När nätverket appliceras på en träningsvektor kan avvikelsen (eng. error) från det förväntade värdet beräknas genom ekvation 4.10. Z är mängden neuroner i utdatalagret och t_z det förväntade värdet för neuron z. Den neuron som motsvarar den rätta klassen förväntas ge värdet 1, alla andra värdet 0. Den hopräknade avvikelse är ett mått på hur vällärt nätverket är och används för att träna det.

$$e(x) = \frac{1}{2}\sum_{z=1}^{Z} (t_z - g_z(x))^2 \quad (4.10)$$

Ett flerlagersnätverk tränas enklast genom en metod kallad bakåtspridning (eng. backpropagation). Avvikelsen för varje neuron i det sista lagret används för att justera dess vikter med målet att minimera avvikelsen. Bakåtspridningen syftar på att det föregående lagret i sin tur justerar sina neuroners vikter enligt avvikelsen som fördelas proportionerligt enligt de olika neuronernas aktiveringsgrad. Ju aktivare en neuron är desto mer justeras dess relaterade vikter.

Nätverket tränas upprepade gånger med data ut träningsmängden tills det är tillräckligt vällärt. Olika kriterier kan användas för att bestämma när träningen bör avslutas, t.ex. träning i ett fixt antal cykler eller tills felfrekvensen på testmängden inte längre blir lägre. (Tarca, s. 3-4; Witten, s. 223-233; Duda, s. 282-296)

4.5 Tillämpning: Cancerdiagnostisering med neurala nätverk

Cancertumörer diagnostiseras vanligen genom att prover studeras i mikroskop eller utsätts för olika kemiska tester, analysmetoder som går under beteckningarna histologi och immunohistokemi. Det finns dock vissa tumörtyper som kan vara svåra att identifiera på dessa vis. Khan m.fl. presenterar en metod för att med hjälp av neurala nätverk kunna klassificera svåridentifierade tumörer utgående från genexpressionsdata.

I deras experiment används den i vissa fall svårklassificerade tumörgruppen barndoms-SRBCT (liten rund blåcellstumör, eng. small round blue-cell tumour), som kan klassificeras i de fyra olika diagnostiska kategorierna neuroblastom (NB), rhabdomyosarkom (RMS), icke-Hodgkinlymfom (NHL) och Ewing-tumörfamiljen (EWS). Datamängden som Khans grupp hade att tillgå innehöll 88 prover, varav 63 användes för träning och validering samt 25 för den slutgiltiga testningen. Varje prov innehöll från början 6567 geners expressionsnivåer.

Först filtrerades de minst differentierade generna bort statistiskt (se 2.1.1), så att 2308 st återstod. Sedan analyserades dessa gener med PCA (se 2.2) och de 10 första principalkomponenterna bibehölls för att senare fungera som attribut för det neurala nätverket (se 4.4). Khans grupp valde att använda ett neuralt nätverk utan dolt lager, eftersom mängden träningsdata var så pass liten. Nätverkets fyra utgångar representerar de olika tumörklasserna.

För träningsprocessen användes korsvalidering (se 5.1) så att de 63 proverna slumpmässigt delades upp i tre partitioner, två för träning och en för validering. För varje uppdelning som gjordes användes alla tre partitioner i tur och ordning som valideringsmängd, vilken behövs för optimeringen som beskrivs i slutet av denna sektion. Partitioneringen och träningen upprepades 1250 gånger så att totalt 3750 nätverk hade tränats till slut. Varje nätverk tränades i 100 iterationer.

För att kombinera alla tränade nätverk användes de 25 prover som i början reserverats för den slutliga testningen. Alla nätverk användes för att klassificera dessa prover och den klass som valdes av flest nätverk blev den slutgiltiga klassen. För diagnostisering är det viktigt att prover som inte hör till någon av de fyra klasserna avfärdas, därför infördes ett statistiskt tröskelvärde. Den Euklidiska distansen i kvadrat mellan medeltalet för alla röster och det ideala utfallet beräknades för alla prover. Detta användes sedan för att beräkna ett konfidensintervall för konfidensgraden 95% (se 5.2). Endast prover innanför detta intervall accepterades till klassen.

Kahns grupp testade även att variera olika geners expressionsnivåer under upprepade försök och studera hur felfrekvensen varierade för nätverkens valideringsmängder (jfr. 2.1.2). På så vis kunde de identifiera vilka geners expression som korrelerar med en viss cancertyp. För att optimera klassificeraren togs irrelevanta gener successivt bort ur träningsmängden och till slut återstod

endast 96 gener med vilka tumörerna kunde klassificeras med 100% säkerhet.

Precis som i fallet i sektion 3.3, kan även här kunskapen om vilka gener som är signifikanta vara en värdefull biprodukt för att undersöka de bakomliggande biologiska orsakerna till en cancerform. Detta är någonting som de traditionella diagnostiseringsmetoderna inte förmår. (Khan)

5. Evaluering

När man använder maskininlärning för att utvinna kunskap ur data är det viktigt att också försöka utvärdera det resultat man får. Tarca m.fl. påpekar att kvaliteten och trovärdigheten hos de resultat man får från dessa experiment kan variera mycket beroende på användarens sakkunskap. I detta kapitel beskrivs kort hur tillförlitligheten hos den utvunna kunskapen kan undersökas och vad som kan göras för att förbättra den. (Witten, s. 143-146; Tarca, s. 10)

5.1 Korsvalidering

I sektion 4.1 diskuteras grunderna för hur en datamängd bör delas upp i tränings- och testmängder innan man kan göra någon övervakad inlärning. Korsvalidering (eng. cross-validation), som är nödvändigt när endast en väldigt liten tränings-mängd finns tillgänglig, är en fortsättning på detta.

En klassificerare som är tränad på en liten mängd data kommer generellt att fungera sämre än en som är tränad på mer data. Korsvalideringen är ett sätt att maximera användningen av den data man har, trots att den måste delas upp i tränings, test och kanske valideringsmängder.

Först delas hela datamängden slumpmässigt upp i *n* olika partitioner. Av dessa används ett visst antal för testning och eventuellt validering och resten för träning. Typiskt kan värdet på *n* vara 3 så att två partitioner används för träning och en för testning. Träningsprocessen upprepas så att alla kombinationer av partitioner i de olika mängderna används. Sedan kombineras resultaten, exempelvis genom röstning bland de lärda klassificerarna, som i experimentet som presenteras i sektion 4.5. (Witten, s. 143-146, 149-151)

5.2 Konfidensintervall

Att mäta prestandan på en klassificerare i felfrekvensen baserat på en testmängd är inte alltid så tillförlitligt eller informativt. Konfidensgraden är ett statistiskt mått på variationen eller tillförlitligheten i en viss uppmätt frekvens.

Klassificeringsprocessen kan ses som en Bernoulli-process, som producerar en stokastisk serie av lyckade och misslyckade utfall. Processen har en verklig men okänd sannolikhet för framgång *p*. Den uppmätta framgångsfrekvensen *f* beräknas av antalet korrekta klassificeringar per totala antalet försök *N*. Man bestämmer den konfidensgrad man är intresserad av, t.ex. 90%. Sedan beräknas det konfidensintervall inom vilket den verkliga sannolikheten för framgång *p* ligger med den angivna graden av säkerhet.

$$b = \left(f + \frac{z^2}{2N} \pm z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right) / \left(1 + \frac{z^2}{N} \right) \quad (5.1)$$

Ekvation 5.1 definierar de relativa gränserna för konfidensintervallet för en given konfidensgrad. Större värden på *N* ger smalare konfidensintervall. En 90%:s konfidensgrad ger *z* värdet 1,65, dessa värden finns ofta angivna i tabeller för praktiskt bruk. (Witten, s. 146-149)

5.3 Expertevaluering

När man evaluerar resultat från ett maskininlärningsexperiment är det viktigt att även ta i beaktande den domän datan kommer ifrån. Experter behövs för att kunna utvärdera om den kunskap man har fått fram är vettig och praktiskt nyttig, huruvida den tidigare vetenskapliga förståelsen på området understöder den nya kunskapen.

Denna avhandling fokuserar på tillämpningar inom molekylärbiologins domäner. De exempel på tillämpningar som ges i sektionerna 2.3, 3.3 och 4.5, visar alla hur någon form av kunskap fås fram som sedan kan analyseras vidare av experter på området. Metoder för visualisering är ett tydligt exempel på hjälpmedel för forskning. Men även en klassificerare som diagnostiserar cancer kan bidra med viktig kunskap om vilka gener som är inblandade, vilket i sin tur ger klassificeraren större trovärdighet. (Valfar, s. 15)

6. Diskussion

Denna avhandling har presenterat olika viktiga tillvägagångssätt för automatisk dataanalys med hjälp av maskininlärning, med inriktning på hur den kan tillämpas för analys av genexpression. Maskininlärning är ett väldigt omfattande område, vad gäller mängden tillämpningsområden så väl som utbudet på metoder. De metoder som har presenterats här utgör bara en ytskrapning, men är ändå valda för att så bra som möjligt illustrera centrala tankegångar inom disciplinen.

När man skall analysera data är det ofta klokt att börja med de simplaste metoderna för att sedan prova sig fram. Genomgången här har varit strukturerad enligt det tankesättet. Det finns dock många andra möjliga sätt att angripa ämnet på, beroende på intresseområde och detaljnivå.

Den avancerade analysen av genexpression och annan biologisk data erbjuder en ny inblick i livets underliggande mekanismer som inte tidigare har varit möjlig. Andra exempel på hur maskininlärning används inom molekylärbiologin är bl.a. jämförelse av DNA-sekvenser för att studera evolutionen och att utgående från en gen förutspå proteinstrukturen och på så vis förstå dess funktion.

Denna översikt fungerar som en introduktion till maskininlärning, men i viss mån även bioinformatik där tillämpningen av maskininlärning utgör ett viktigt delområde.

7. Litteraturförteckning

1. Pierre Baldi & G. Wesley Hatfield; DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling; 2002
2. Pierre Baldi & Soren Brunak; Bioinformatics: The Machine Learning Approach, 2nd ed; 2001
3. Xavier Castells, Juan Miguel García-Gómez, Alberto Navarro, Juan José Acebes, Óscar Godino, Susana Boluda, Anna Barceló, Robles Montserrat, Joaquín Ariño & Carles Arús; Automated Brain Tumor Biopsy Prediction Using Single-labeling cDNA Microarrays-based Gene Expression Profiling
4. David L. Nelson & Michael M. Cox; Lehninger Principles of Biochemistry, 5th ed; 2008
5. Richard O. Duda, Peter E. Hart & David G. Stork; Pattern Classification, 2nd ed; 2001
6. Javed Khan, Jun S. Wei, Markus Ringnér, Lao H. Saal, Marc Ladanyi, Frank Westermann, Frank Berthold, Manfred Schwab, Cristina R. Antonescu, Carsten Peterson & Paul S. Meltzer; Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artifical Neural Networks; 2001
7. P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Perez & V. Robles; Machine Learning in Bioinformatics; 2006
8. S. Ott, S. Imoto & S. Miyano; Finding Optimal Models for Small Gene Networks; 2004
9. Dov Stekel; Microarray Bioinformatics; 2003
10. Adi L Tarca, Vincent J. Carey, Xue-wen Chen, Roberto Romero & Sorin Draghici; Machine Learning and Its Applications to Biology; 2007
11. Faramarz Valafar; Pattern Recognition Techniques in Microarray Data Analysis: A Survey; 2002
12. Ian H. Witten & Eibe Frank; Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed; 2005