Kan man lära datorer att läsa?

N I K L A S L U N D B O R G

KTH Datavetenskap
och kommunikation

Examensarbete
Stockholm, Sverige 2011

Kan man lära datorer att läsa?

N I K L A S L U N D B O R G

Examensarbete i medieteknik om 15 högskolepoäng
vid Programmet för medieteknik
Kungliga Tekniska Högskolan år 2011
Handledare på CSC var Johan Boye
Examinator var Mads Dam

URL: www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2011/
lundborg_niklas_K11060.pdf

Kungliga tekniska högskolan
Skolan för datavetenskap och kommunikation

KTH CSC
100 44 Stockholm

URL: www.kth.se/csc

Referat

En undersökning av olika typer av klustringsalgoritmer har genomförts. Undersökningen behandlar en intressant probleminstans, teckenigenkänning. Det som är intressant med att använda klustring för teckenigenkänning är att klustring är en metod som bygger på oövervakad inlärning och det som utmärker oövervakade inlärningsmetoder är att sådana metoder ej känner till vilken klass ett objekt tillhör i upplärningsfasen. Detta skiljer sig markant från vanliga teckenigenkänningsmetoder, där man ofta tränar genom att titta vilka egenskaper som finns för varje klass av objekt.

De klustringsalgoritmer som undersökts är *K-means, Xmeans, hiearkisk klustring* och till dessa har olika avståndsberäkningar använts , *Euklidskt avstånd, Cosinusmått och Manhattan avstånd.*

Syftet med undersökningen är att ta reda på om det går att använda klustringsalgoritmer för att utföra teckenigenkänning och undersöka vilka förutsättningar som behövs för att lyckas.

Slutsatsen är att det går att använda klustring i detta syfte, om man har möjlighet att påverka vilka parametrar som används. I denna undersökning så har teckenparametrar som är mer direkt relaterade till hur bilden på tecknet ser ut visat sig fungera bra. För att mer nogrannt avgöra vad som utmärker en bra parameter behövs en mer omfattande undersökning.

Abstract

Can computers be taught to read?

An evaluation on different sorts of cluster analysis have been performed. To do this evaluation an interesting problem-set will be used, character recognition. This is interesting because cluster analysis is a method built on unsupervised learning. What distinguish unsupervised learning from other methods is that it have no knowledge about what class a object belong to when evaluating. This is an important diffrence from other character recognition techniques, where they often learn by looking at what features a specific character has.

The cluster algorithms evaluated are *K-means, Xmeans* and *Hierarchical clustering*, and for these methods the following distance measures have been used *Euclidian distance, Cosine similarity* and *Manhattan distance*.

The objective of this evaluation is to find out if its possible to use cluster analysis to do character recognition and to evaluate the conditions needed to do this succesfully.

The main conclusion is that it indeed is possible to use cluster analysis to perform character recognition, if you have the possibility to affect the parameters used. In this evaluation it has been found that parameters that map more directly to the picture representation of the characters have performed better. To determine more exact what a good parameter is a more extensive analysis is needed.

Innehåll

Kapitel 1

Introduktion

1.1 Inledning

Att lära en dator läsa handskriven text är något som är mycket användbart inom många områden. I denna uppsats kommer teckenigenkänning med hjälp av klustringsalgoritmer att undersökas. Klustring är en metod för att statistiskt klassifiera stora datamängder i så kallade kluster. Det som utmärker klustring och det som göra denna uppsats intressant är att klustring bygger på *unsupervised learning* [12], oövervakad inlärning.

Oövervakad inlärning innebär att i inlärningsfasen ges ingen kännedom om vilken klass en datapunkt bör klassificeras som, vilket är en precis motsats till *supervised learning* [12],övervakad inlärning som många andra teckenigenkänningsmetoder bygger på. Detta betyder att det är klustringsmetodens uppgift att hitta dessa klasser. Undersökningen kommer gå ut på att undersöka och utvärdera flera olika klustringsalgoritmer och till dem prova olika sätt att mäta skillnader mellan olika teckenparametrar.

1.2 Syfte

Huvudsyftet med detta projekt är att undersöka huruvida man kan göra teckenigenkänning med hjälp av klustringsalgoritmer med en tillräcklig hög precision för att det ska bli användbart. Det är i huvudsak två klustringsalgoritmer som kommer undersökas, *K-Means Clustering* och *Hierarkisk klustring*, men även andra algoritmer ska undersökas. Metoderna ska jämföras med varandra och se i vilka situationer de fungerar bäst. Det kommer också att undersökas om man kan förbättra resultatet genom att påverka någon teckenparameter t.ex genom att ta bort den vid klustringen. Detta för att för försöka dra slutsatser om vilka parametrar som betyder mest.

Undersökningar ska utföras på vad som händer om man ändrar avståndsberäkningen. De avståndsmätningar som har använts är *euklidiskt avstånd*, *cosniusmått* och *manhattanavstånd*. Undersökningen ska även ta upp hur resultatet blir om man ändrar antalet kluster till mindre eller större än antalet tecken.

Undersökningen kommer också försöka dra slutsatser om varför någon metod blir fel, om det t ex kan bero på att vissa bokstäver är för lika andra bokstäver för att kunna

skilja på dem. Undersökningen ska också försöka dra slutsatser om vad som ger dåliga respektive bra resultat.

1.3 Bakgrund

Det finns många olika metoder för att lära datorer saker och ting och vetenskapen har ett samlat namn, maskininlärning. Maskininlärning är en gren av artificiell intelligens som utnyttjar statistiska resonemang för att dra egna slutsatser från empirisk data. För att lära en dator så måste man först träna den. Träningen går ut på att analysera stora mängder data och sedan ställa upp statistiska modeller om förhållandet mellan olika parametrar i datamängden. Det finns två väsentligt olika metoder vid maskininlärning, *supervised learning*(*övervakad inlärning*) och *unsupervised learning*,(*oövervakad inlärning*).

Vid övervakad inlärning arbetar metoden med data där man känner till vilken klass(vilket tecken) den hör till. Syftet är att bygga en funktion som efter inlärning kan mappa okänd data till någon av dessa (kända) klasser. Styrkan med övervakad inlärning är att den efter en tillräcklig inlärning kan ge mycket bra resultat.

Oövervakad inlärning är väsentligen annorlunda, där finns vid inlärning ingen kännedom om vad datan har för klasstillhörighet och det är metodens uppgift att finna dessa klasser. Styrkan med oövervakad inlärning är att metoden kan finna samband mellan data som annars inte hittas.

Denna undersökning går ut på att ta reda på man kan utnyttja en oövervakad inlärningsmetod för att åstadkomma teckenigenkänning. Ett samlingsnamn för metoden som kommer att användas är *klustring*. Det finns flera olika klustringsmetoder och denna undersökning ska titta några av de mest kända metoderna. Det som gör det intressant att göra teckenigenkänning på just bokstäver och siffror med hjälp av en oövervakad inlärningsmetod är att det går att verifiera resultaten eftersom man känner till det förväntade resultatet. Det är även intressant att se om våra bokstäver och siffror har andra intressanta samband.

Ett problem som uppstår när man använder oövervakad inlärning är att klassificeringarna som bildas inte nödvändigvis är de klassificeringar som de borde vara. Det ligger i oövervakade inlärningsmetoders natur att just hitta sådana okända klassificeringar. Detta ger upphov till ett problem att evaluera resultatet även om man känner till det förväntade resultatet. I denna undersökning ska tecken som är lika klassificeras till samma klass. Det är alltså känt vilka tecken som bör hamna tillsammans, men om det visar sig att det bildas klasser med en blandning av tecken så behöver detta resultat inte vara ett dåligt resultat, utan det betyder att det finns en struktur på de tecken som är i samma klass som för människor är svår att förstå.

1.4 Avståndsmätning

För att utföra klustring så behöver man kunna mäta avstånd mellan datapunkterna. I denna undersökning så kommer två färdiga datamängder [6](Se avsnitt 1.3.3 för noggrann beskrivning) användas som har 16 parametrar respektive 64 parametrar. Avståndet mellan två punkter kommer att bero på alla dessa parametrar. För att göra avståndsberäkningen så kommer olika metoder att användas för att jämföra vilken metod som ger bäst resultat.

1.4.1 Euklidiskt avstånd

Det euklidiska avståndet innebär i två dimensioner precis samma sak som 'raka vägen' eller fågelvägen.
Mer formellt definierat som: Låt $P = \{p_1, p_2, ..., p_n\}$ och $Q = \{q_1, q_2, ..., q_n\}$ vara punkter i n dimensioner. $d_{Euk}(Q, P) = d_{Euk}(P, Q)$(Kommutativt) står för avståndet mellan p och q.
$d_{Euklidisk}(p, q) = d_{Euklidisk}(q, p) =$

$$\sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + ... + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{1.1}$$

1.4.2 Cosinusmått

Cosine-similarity översätts oftast som *cosinusmått* och är ett mått på likhet. Punkterna ses som (normerade) vektorer och cosinusmåttet är cosinus av vinkeln mellan dem. Detta ger följande egenskaper:

- Om vektorerna är lika blir vinkeln mellan dem 0 vilket ger $cos(0) = 1$,total likhet.
- För alla andra vinklar så blir cosinusmåttet mindre än 1.
- Vektorernas längd spelar ingen roll.

För de andra avståndsmåtten så innebär ett större avstånd mindre likhet. För detta mått är det tvärtom och därför inverteras måttet innan användning.
Mer formellt definierat som:
Låt $P = \{p_1, p_2, ..., p_n\}$ och $Q = \{q_1, q_2, ..., q_n\}$ vara punkter i n dimensioner. $s_{cos}(P, Q) = s_{cos}(Q, P)$ står för avståndet mellan P och Q.

Låt u och v vara de normerade vektorerna som spänns upp av $u = norm(\overrightarrow{\overline{0}P})$ respektive $v = norm(\overrightarrow{\overline{0}Q})$ där $\overline{0}$ är origo.

$$s_{cos}(P, Q) = s_{cos}(u, v) = \frac{u \cdot v}{|u||v|} \tag{1.2}$$

Som angivet ovan så måste måttet inverteras innan användning
($d_{cosinv}(P, Q) = 1 - s_{cos}(P, Q)$). Se även avsnittet Problem som upstått(Avsnitt 4.1.1).

1.4.3 Manhattanavstånd

Namnet Manhattanavstånd kommer från en jämförelse med hur långt man måste gå mellan två höghus i Manhattan, det går inte att gå raka vägen utan man måste följa de vinkelräta gatorna. Sträckan mellan två punkter är således skillnaden i x-led adderat med skillnaden i y-led.
Mer formellt:
Låt $P = \{p_1, p_2, ..., p_n\}$ och $Q = \{q_1, q_2, ..., q_n\}$ vara punkter i n dimensioner.
$d_{Manh}(P, Q) = d_{Manh}(Q, P)$ står för avståndet mellan P och Q.

$$d_{Manh}(P, Q) = \sum_{i=1}^{n} |q_i - p_i| \tag{1.3}$$

1.5 Algoritmer

I detta avsnitt kommer de algoritmer som har undersökts att definieras och förklaras. De algoritmer som används är k-means clustering, hierarkisk klustring och Xmeans.

1.5.1 K-means

K-means clustering är en klustringsalgoritm för att partionera datamängder i k olika kluster. Algoritmen har två övergripande steg.

- 1. Varje objekt(datapunkt) $o \in datamängd$ placeras till det kluster som är (enligt någon avståndsmätning) närmast.
- 2. Centroiden för varje kluster uppdateras med medelvärdet av de punkter som blivit tilldelat det klustret.

Centroiden för varje kluster är initialt slumpmässigt valda punkter från datamängden. När steg två i algoritmen inte ändrar värdet på centerpunkten eller när den har gjort ett fixt antal iterationer så terminerar algoritmen. Oftast är det den tidigare som gör att algoritmen terminerar.

1.5.2 X-means

XMeans är en utökning av *Kmeans* med skillnaden att vid varje iteration så försöker den förbättra resultatet genom att söka igenom alla klustringscentroider och avgöra om det är bättre att dela upp denna i olika kluster. I denna algoritm så anger man ett intervall av önskade antal kluster, och algoritmen avgör hur många kluster som blir bäst.

1.5.3 Heirarkisk klustring

Hierarkisk klustring skapar en hierarki av kluster i form av ett träd. Roten av noden är hela datamängden, och sedan så delas datamängden upp i grenar och löven representerar de ensamma datapunkterna. Det finns två olika varianter av algoritmen. En där man

börjar med hela mängden data och delar upp den i två delar, och fortsätter med att dela upp dessa delar i fler delar tills man har önskad mängd kluster. I den andra varianten så börjar man från varje enskild datapunkt och försöker hitta närliggande datapunkter för att på så vis bygga upp hierarkin. Den förstnämnda kallas *divisive*, alltså *uppdelande*. Den senare kallas *agglomerative*, vilket betyder ungefär *ihopsmältande* eller *ihopgyttrande*.

Se även avsnittet Problem som uppstått - Minne (Avsnitt 4.1.2).

Kapitel 2

Metod

I denna sektion kommer följande områden tas upp:

- Vilka olika sorters undersökningar som har gjorts.
- En beskrivning av de datamängder som använts.
- Programvaran som använts för undersökningarna.
- Hur undersökningarna har utvärderats.

2.1 Undersökningar

Nedan i tabell 2.1 är en sammanställning av samtliga undersökningar som har gjorts.

Test nummer	Metod	Antal kluster	Avståndsberäkning
B1	K-Means	26	Euklidiskt
B2	K-Means	26	Cosinusmått
B3	K-Means	26	Manhattan
B4	Hierarkisk	26	-
B5	Xmeans	10-40	Euklidiskt
B6	Xmeans	24-28	Euklidiskt
S1	K-Means	10	Euklidiskt
S2	K-Means	10	Cosniusmått
S3	K-Means	10	Manhattan
S4	Hierarkisk	10	Euklidskt
S5	Hierarkisk	10	Manhattan
S6	Xmeans	6-12	Euklidiskt
S6	Xmeans	6-12	Manhattan

Tabell 2.1. Undersökningar som ska göras. Test med B är test med bokstavsdatamängden, och test med S är med sifferdatamängden.

För varje test så kommer resultatet visualiseras och bedömas.

Ytterliggare en undersökning ska göras som behandlar teckenparametrarna. Det ska undersökas hur resultatet påverkas genom att successivt ta bort en parameter i taget, för att se om någon av dessa ger ett bättre resultat. Resultaten förutsätts bli ungefär lika oavsett vilken klustringsmetod som används, men det kan däremot skilja hur avståndsberäkningen utfaller.

2.2 Datamängd

För att göra denna undersökning så behövs en datamängd. UCI [5] är ett arkiv med datamängder för testning och användning av maskininlärningsalgoritmer. Från detta arkiv har två olika typer av datamängder används.

2.2.1 Bokstäver

Från arkivet ovan nämnt kommer ett datamängdspaket[6] användas. Datamängden är baserad på det engelska alfabetet och innehåller därmed 26 olika tecken. Datamängden består av tecken från 20 stycken olika fonter som sedan har blivit slumpmässigt blivit förvrängda(Se figur 2.1 nedan).

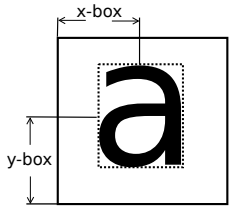
Figur 2.1. Exempel på hur de förvrängda bokstäverna kan se ut. Illustration lånad från Letter recognition using Holland-style adaptive classifiers[10]. Detta är en bild från de riktiga bokstäverna som datamängden är skapade från.

Totalt i datamängden finns 20000 förvrängda tecken. Varje objekt i datamängden har fått 16 st numeriska värden i intervallet 0-15. Dessa värden är framtagna i en annan undersökning [10] med hjälp av olika metoder som t ex medelvärde av pixlar i x-led. Se tabell 2.2 nedan för en mer noggrann beskrivning.

Attribute number	Attribute identifier	Attribute Information
1	letter	capital letter (26 values from A to Z)
2	x-box	horizontal position of box (integer)
3	y-box	vertical position of box (integer)
4	width	width of box (integer)
5	high	height of box (integer)
6	onpix	total amount on pixels (integer)
7	x-bar	mean x of on pixels in box (integer)
8	y-bar	mean y of on pixels in box (integer)
9	x2bar	mean x variance (integer)
10	y2bar	mean y variance (integer)
11	xybar	mean x y correlation (integer)
12	x2ybr	mean of x * x * y (integer)
13	xy2br	mean of x * y * y (integer)
14	x-ege	mean edge count left to right (integer)
15	xegvy	correlation of x-ege with y (integer)
16	y-ege	mean edge count bottom to top (integer)
17	yegvx	correlation of y-ege with x (integer)

Tabell 2.2. Datamängdens parametrar

För att exemplifiera hur man tar fram parametrarna så visas nedan en bild som visar parameter 2 respektive 3 i tabell 2.2. Den streckade rektangelen motsvarar den box som används i tabell 2.2 ovan vid många av parametrarna. Parameter 2 motsvarar sträckan från vänster kant till mitten av den streckade rektangeln, räknat i antal pixlar. Parameter 3 motsvarar vertikal position och räknas från botten.



Figur 2.2. Denna bild visar vad parameter 2 och 3 från tabell 2.2 representerar.

2.2.2 Siffror

Från samma arkiv finns en uppsättning siffror[7]. Datamängden är uppbyggd av handskrivna siffror $[0-9]$ från totalt 43 olika individer. Det finns två versioner av datamängden. Den första bygger på 32×32 stora bitmaps[11] med booleska värden$(0, 1$ där 1 är färgad pixel, 0 en ofärgad pixel). Detta ger upphov till en datamängd med $1024 = 32 \times 32$ booleska attribut vilket inte är praktiskt när datamängden ska användas. Av denna anledning gjorde de en andra version av datamängden som istället har $64 = 8 \times 8$ attribut. Detta är gjort genom att dela upp bitmappen i block om 4×4 och sedan räkna anatalet pixlar i varje block. Något som är intressant med denna datamängd är att den är uppdelad i en träningsdel och en testdel och det är olika grupper av individer som har gjort respektive del. 30 individer stod för träningsdelen och restrerande 13 skapade testdelen.

```
00000000000111000000000000000000
00000000001111111000000000000000
00000000011111111110000000000000
00000000011111111111100000000000
00000000011111111111100000000000
00000000011111111111110000000000
00000000011111111111111100000000
00000000011111000011111100000000
00000000011111000000111110000000
00000000011111000000111110000000
00000000000111100000111110000000
00000000000111100000111110000000
00000000000111100001111110000000
00000000000111110111111110000000
00000000000111111111111100000000
00000000000011111111110000000000
00000000000011111111100000000000
00000000000011111111000000000000
00000000000011111111000000000000
00000000011111111111100000000000
00000000111111111111110000000000
00000000111111000111111000000000
00000011111111000001111110000000
00000001111111000000111111100000
00000001111110000000011111100000
00000001111110000000011111100000
00000000111110000000011111100000
00000000111110000000011111100000
00000000011111000001111111100000
00000000011111111111111110000000
00000000011111111111111100000000
00000000011111111111110000000000
00000000001111111110100000000000
```

Figur 2.3. Visar hur siffran 8:a ur datamängden såg ut innan formatering.

2.3 Programvara

2.3.1 WEKA

För att kunna utföra denna undersökning kommer open-source programvara att användas. Programmet heter WEKA[8] och är ett sofistikerat programpaket för olika tillämpningar av maskininlärningsalgoritmer avsett för data-mining[9]. WEKA är utvecklad av maskininlärningsgruppen på *University of Waikato*.
WEKA tillhandahåller många verktyg som detta arbete kommer ha användning av, för att nämna några viktiga detaljer:

- Relevanta klustringsalgortimer
- Filter för hantering av datamängden, t ex normalisering eller ta bort någon parameter.
- Visualisering
- Dataanalys
- GUI
- Öppen källkod, vid behov av förändringar

2.4 Utvärdering

För att kunna bedöma hur bra en klustring blev så har en 12-gradig skala använts. Denna 12-gradiga skala har använts på grund av den mängd data som man får ut när man klustrar 26 olika tecken är mycket stor och svår att få intuitiv förståelse för. Denna skala är således ett försök att på ett enkelt sätt förmedla hur bra respektive dåligt en klustringsalgoritm har fungerat. Ett svårighet när man ska evaluera klustringsalgoritmer är att om ett tecken har hamnat i ett kluster så finns inget sätt att avgöra om den är i rätt eller fel kluster. Det enda man kan säga är att viss mängd av alla tecken av en speciell typ har hamnat i något kluster.

Nedan i tabell 2.3 så finns en förklaring till vad varje värde i skalan betyder. Varje kluster kommer utvärderas med denna skala, och sättas samman i ett histogram där man ser hur många kluster som fick respektive värde. För att få en förståelse för vad klustringsmetoden gör för fel så kommer även en lista med tecken som har hamnat i samma kluster att visas. Att tänka på när man läser av histogrammen är att resultat 12 går endast att uppnå om ett teckens hela datamängd hamnar i samma kluster, samtidigt som ingen annan bokstav får hamna i det klustret. Att uppnå värde 12 är således väldigt svårt, inte ens de bästa metoderna klarar detta.

Det finns andra sätt att evaluera klusteralgoritmer [2], men dessa är tyvärr inte tillämpbara i denna undersökning. Att de traditionella evalueringsmetoderna inte fungerar beror på att de bygger på att evaluera klustringar där man inte känner till de korrekta resultaten, medans i denna undersökning är detta känt. Denna typ av evaluering går

bland annat ut på att avgöra hur långt olika klustermedelpunkter ligger ifrån varandra, och i denna undersökning kan ett perfekt resultat(alla bokstäver hamnar i ett eget kluster) ge ett dåligt resultat med en sådan evaluering. Detta på grund av att den datamängdens korrekta klasstillhörighet inte stämmer överens med vad som anses vara ett bra kluster. Om så är fallet så innebär det att klustermedelpunkterna ligger nära varandra och detta är enligt andra evalueringsmetoder dåligt.
För att undvika redundans i tabellen så införs följande begrepp:

- $SINGLE(x)$ En bokstav har $x\%$ av sin totala datamängd i detta kluster.
- $MULT(x)$ Flera bokstäver har $x\%$ av deras totala datamängd i detta kluster.
- $ALLOTHERS(x)$ Alla andra bokstäver har mindre än $x\%$ av sin totala datamängd i detta kluster.
- $ALL(x)$ Alla bokstäver har mindre än $x\%$ av sin totala datamängd i detta kluster.

Värde	Förklaring	
12	$SINGLE(100)$	$ALLOTHERS(0)$
11	$SINGLE(100)$	$ALLOTHERS(10)$
10	$SINGLE(100)$	$ALLOTHERS(30)$
9	$SINGLE(70)$	$ALLOTHERS(10)$
8	$SINGLE(70)$	$ALLOTHERS(30)$
7	$SINGLE(50)$	$ALLOTHERS(10)$
6	$SINGLE(50)$	$ALLOTHERS(30)$
5	$SINGLE(30)$	$ALLOTHERS(10)$
4	$MULT(50)$	$ALLOTHERS(10)$
3	$MULT(50)$	$ALLOTHERS(30)$
2	$ALL(30)$	
1	$ALL(10)$	

Tabell 2.3. 12-gradig skala för att evaulera en klustringsmetod

Kapitel 3

Resultat

I denna del finns en sammanställning av samtliga resultat. Sammanställningen är gjord för att kunna snabbt avgöra vilken metod med tillhörande avståndsmätning som har fungerat bäst. Värdet i kolumnen längst till höger i tabell 3.1 är ett medelvärde av en metods alla klusterevalueringar, där evalueringen är gjord med den 12-gradiga skalan omnämnd tidigare. I Bilaga A(Data) så finns mer data om varje test, därbland visualiseringar och bokstäver som blivit dömda som lika.

Testnummer	Metod	Antal kluster	Avståndsberäkning	EVAL12-Medelvärde
B1	K-Means	26	Euklidiskt	2.423
B2	K-Means	26	Cosinusmått	2.153
B3	K-Means	26	Manhattan	1.961
B4	Hierarkisk	26	-	-
B5	Xmeans	10-40	Euklidiskt	2.35
B6	Xmeans	24-28	Euklidiskt	1.964
S1	K-Means	10	Euklidiskt	6.0
S2	K-Means	10	Cosniusmått	6.4
S3	K-Means	10	Manhattan	7.3
S4	Hierarkisk	10	Euklidskt	4.0
S5	Hierarkisk	10	Manhattan	3.9
S6	Xmeans	6-12	Euklidiskt	5.916
S6	Xmeans	6-12	Manhattan	6.166

Tabell 3.1. En sammanställning av EVAL12-medelvärden för alla tester, se datasektionen för mer utförlig redovisning.

Kapitel 4

Diskussion

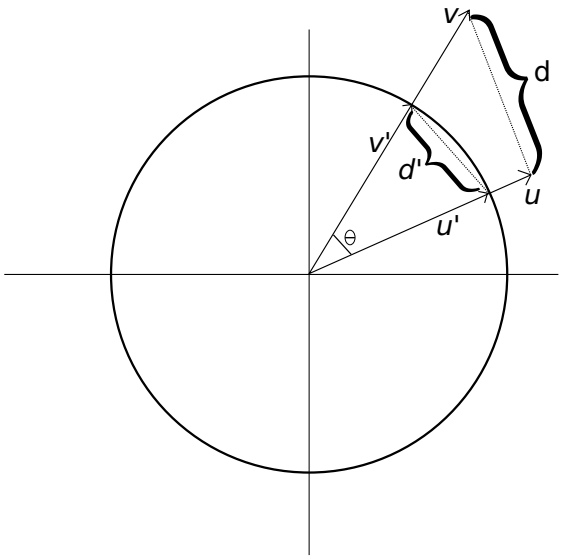
I denna sektion kommer resultaten analyseras och förklaras i den mån det är möjligt. Även problem som har uppstått kommer att förklaras och visa hur dessa har blivit lösta.

4.1 Problem som uppstått

4.1.1 Cosinusmåttet

Weka tillhandahåller alla avståndsberäkningar som behövs, förutom cosinusmåttet. Alla de avståndsfunktioner som finns implementerade i WEKA utnyttjar en differans för att beräknas, och således är deras gränsnitt anpassat efter det. Detta medför problem om man vill implementera cosinusmåttet(Se Bakgrund för beskriving), eftersom man behöver vektorvärderna. Nedan visas en bild (Figur 4.1) som behövs för att förstå situationen. Givet två vektorer u och v, så kan Weka beräkna det euklidska avståndet d. Om man normaliserar vektorerna $(u \to u', v \to v')$innan weka gör dessa beräkningar så fås istället avståndet d'. Med hjälp av d' kan man få fram cosinusmåttet, eller som nämts i bakgrunden det inverterade cosinusmåttet.

$$
\begin{aligned}
d' &= 2rsin\left(\frac{\theta}{2}\right) \\
d' &= 2 \pm \sqrt{1 - cos^2(\theta)} \text{ använder ekvation B.5} \\
\frac{d'}{2} &= \pm\sqrt{1 - \frac{1 + cos(\theta)}{2}} \text{ använder ekvation B.6} \\
\frac{d'^2}{4} &= \frac{1 - cos(\theta)}{2} \\
\frac{d'^2}{2} &= 1 - cos(\theta) \qquad (4.1)
\end{aligned}
$$



Figur 4.1. Denna bild visar i två dimensioner det euklidiska avståndet d mellan u och v, och om dessa vektorer normaliseras till u' och v' så fås det normaliserade euklidiska avståndet d'. Med detta d' så kan man med ovanstående beräkning få fram cosinusmåttet.

4.1.2 Minne

Det finns ett minnesproblem när man arbetar med stora datamängden. Datamängden med bokstäver innehåller 20000 vektorer av dimension 17. De flesta metoderna fungerar att köra på vanliga datorer, men t ex hierarkisk klustring har en kubisk minneskomplexitet vilket gör det omöjligt även på datorer med mycket minne. Detta har påverkat delar av undersökningen då det inte har gått att bestämma hur bra den metoden fungerar. För referens så räckte minnet till för att utföra hierarkisk klustring på cirka $\frac{1}{4}$ av bokstavsdatamängden och fungerade på hela sifferdatamängden.

4.2 Analys

Resultatet i tabell 3.1 ser man tydligt att ingen av metoderna har fungerat bra för bokstavsdatamängden. Det bästa EVAL12-Medelvärdet som undersökningarna gett är 2.423, vilket är ett mycket dåligt resultat. För att förstå varför det blev dåliga så är ett sätt att titta på vilka bokstäver som har dömts lika enligt respektive metod. Metoden för att bedöma två bokstäver lika är att de har hamnat i samma kluster, och både har mer än 10% av sin totala datamängd i det klustret. De bokstäver som dömts lika för det bästa reslutatet (Test B1) visas nedan. Om man kort reflekterar över vilka bokstäver som har blivit placerade tillsammans så ser man i många fall en likhet. Många tecken med runda former har placerats tillsammans med andra tecken som också har runda former. Liknande resonemang gäller för bokstäver med raka kanter, och även för bokstäver med raka streck som går i ungefär samma riktning, t ex V och Y. Eftersom

B E G S X Z	B D E I J L R S Z
C K U X	B D G O Q R U
V Y	C Q
B D G I K O Q R X	I M N O Q
U W	C G U
F P T	B D E H K L Q R S
C E S Z	F P S T
T V Y	H M N U
F P T V Y	I J

Tabell 4.1. Bokstäver som har dömts lika i Test B1.

metoden inte lyckas att skilja på dessa bokstäver blir felet egentligen större och större för mer inlärningsdata. Klustermedelpunkterna hamnar tätt mot varandra och det blir svårare att skilja på olika bokstäver. Det som orsakar att vissa bokstäver döms som lika är att datamängden är helt enkelt för svår. Svår betyder i denna mening att parametrarna inte passar till klustring. Att detta resultat blev dåligt innebär inte i någon mening att klustringsmetoden för teckenigenkänning är dålig, utan det innebär bara att man måste ha datamängder som passar sig för klustring. En sådan datamängd kommer diskuteras lite längre ner. En annan sak som har gjort det svårt för klustringen är mängden olika tecken, 26 är en ganska stor mängd och vissa tecken är väldigt lika, t ex O≈Q och I≈J.

Det har även utförts ett kortare test för att se om någon av parametrarna gav upphov till bättre eller sämre resultat. Det har gjorts genom att successivt ta bort en parameter i taget för att se om någon påverkade resultatet positivt eller negativt. Resultatet av detta visade enbart negativt, alla parametrar behövs för att få det resultat som fåtts. Det enda intressanta detta gav var att man kan se att vissa parametrar är viktigare än ändra.

På grund av att resultaten blev dåliga infördes en ny datamängd, denna mängd med siffror. Motiven till att byta datamängd är följande:

- Mindre antal olika tecken, problemet blir alltså lättare i den meningen.
- Datamängden har en helt annan typ av parametrar(För detaljer om detta, läs sektionen om datamängd (2.2)) och det är intressant att se hur det påverkar resultatet.

Vid denna undersökning så blev det bästa resultatet(EVAL12-medelvärde)7.3. Ett medelvärde på 7 innebär att i snitt så finns 50% av den totala mängden tecken av samma typ i samma kluster, tillsammans med några få tecken som har fått $< 10\%$ i detta kluster. Om man tittar på histogrammet ser man att EVAL12-Medianen ligger på 9. Ett värde på 9 innebär att samma som 7, men med ändring att 70% av det tecken ligger i samma kluster. Detta innebär att den i praktiken skulle kunna användas med ett ganska gott resultat. Något man bör påminnas om är att metoderna arbetar helt utan kännedom av hur datapunkter bör förhålla sig till varandra, vilket gör detta till en väldigt svår uppgift. Människor använder helt andra metoder för att lära sig läsa och känna igen

tecken, men även för oss tar det lång tid att bli duktiga, även fast vi använder metoder som klassas som *övervakad inlärning*.

4.3 Slutsats

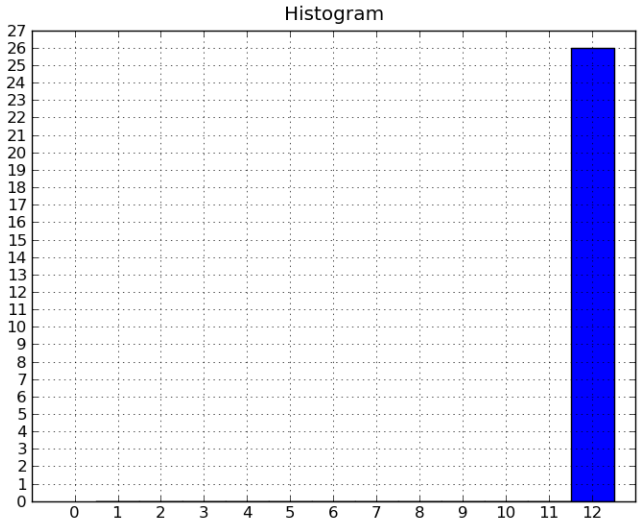
Klustring är en utmärkt metod för att klassificera datamängder. För användning vid teckenigenkänning så bör man först göra en undersökning om vilka parametrar för varje tecken som har högst betydelse, annars får man dåliga resultat. Sifferdatamängden som visade sig fungera bra hade parametrar som var mer direkt relaterade till hur figuren såg ut, och det är antagligen detta som gjorde att klustringsmetoden fungerade bra på denna datamängd. Datamängden med bokstäver innehöll statistisk data om figuren som var anpassade efter den metod som de som skapade datamängden använda, vilket inte tillförde något nyttigt för de metoder som denna undersökning använts. Den klustringsmetod som har fungerat bäst genomgående är *K-means* och vilken avståndsmätning som fungerat bäst har inte gått bestämma. Euklidiskt avståndmätning och manhattanavstånd har båda visat sig fungera bra och cosinusmått har visat sig fungera lite sämre. Hierarkisk klustring har visat sig fungera mycket dåligt i detta syfte. Denna slutsats är grundad i resultatet för siffermängden, där den presterade mycket dåligt jämfört med de andra metoderna. Helt rättvist går det ej att bedöma på grund av att det ej gick att tillämpa hierarkisk klustring på stora datamängder.

Bilaga A

Data

I denna sektion så kommer varje data för varje tests visas. Datan är abstraherad i två eller tre steg, beroende på hur man ser på det. Först så får man utdata från klustermetoden. Den har placerat varje datapunkt i ett kluster. Av denna data bildades en matris där varje rad stod för ett tecken och varje kolumn står för ett kluster. Varje kolumn i denna matris värderades genom den 12-skaliga evalueringsmetoden (Senare förkortat EVAL12). För att få en enkel visualisering av dessa resultat så skapades histogram som visar antalet kluster med varje EVAL12 värde. För att enkelt få en förståelse för vad som har gått fel så visas även tecken som klustermetoden har dömt vara lika varandra.

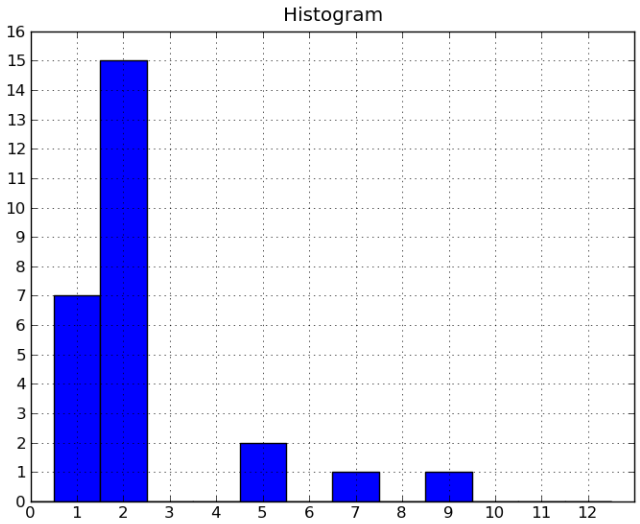
Nedan visar hur ett histogram som har ett perfekt resultat (Bildades genom att klustra med endast en parameter, och det var klassvariablen.)



Figur A.1. Histogram för en perfekt körning $EVAL12$

A.1 Test B1

Metod: *K-means* **Parameter:** 26 **Avståndsmätning:** *Euklidisk* **Datamängd:** *B*



Figur A.2. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

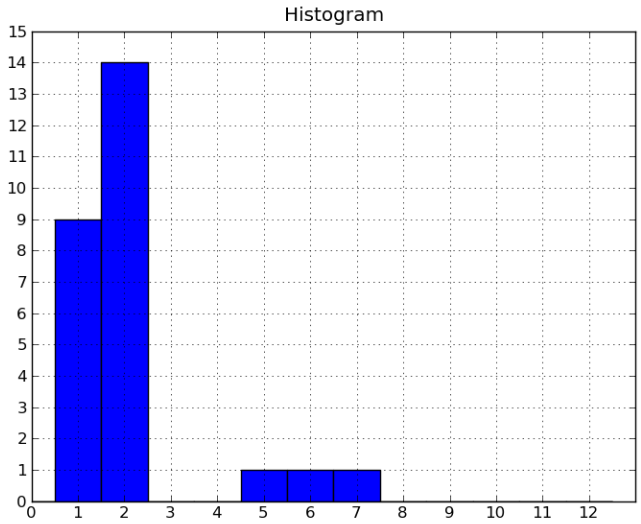
EVAL12-Medelvärde ≈ 2.423

B E G S X Z	B D E I J L R S Z
C K U X	B D G O Q R U
V Y	C Q
B D G I K O Q R X	I M N O Q
U W	C G U
F P T	B D E H K L Q R S
C E S Z	F P S T
T V Y	H M N U
F P T V Y	I J

Tabell A.1. Bokstäver som har dömts lika.

A.2 Test B2

Metod: *K-means* **Parameter:**26 **Avståndsmätning:***Cosniusmått* **Datamängd:***B*



Figur A.3. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

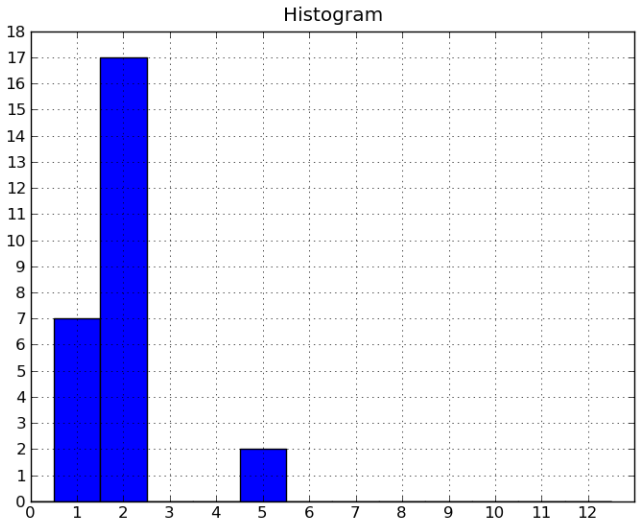
EVAL12-Medelvärde ≈ 2.153

B E Q S X Y	O Q
S X Z	B D G K O Q R X
T V Y	N V W
M N W	B D E I J R S X Z
B D E I J S Z	B D G O Q R
I L	C K U
F T V Y	B C E G S X Z
C E	H N
M W	F T U V Y
H M N	

Tabell A.2. Bokstäver som har dömts lika.

A.3 Test B3

Metod: *K-means* **Parameter:**26 **Avståndsmätning:***Manhattan* **Datamängd:***B*



Figur A.4. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

EVAL12-Medelvärde ≈ 1.961

B S X Z	M W
C D G K O Q R	F P T Y
D N	F P T V W Y
B D R	E K S Z
T V W Y	G W
I J T	B C D E G O Q R S X Z
B D R	B C E G K Q R X Z
I L	H M N U
C U	A L
C K U	

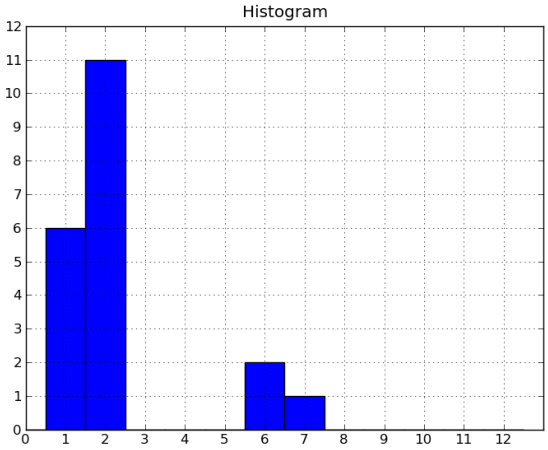
Tabell A.3. Bokstäver som har dömts lika.

A.4 Test B4

Metod: *Hiearkisk* **Parameter:**26 **Avståndsmätning:**- **Datamängd:***B* Se sektion i diskussion om problem. Det visade sig att det ej gick att genomföra hiearkisk klustring på så stora datamängder. Om man reducerade datamängden till $\frac{1}{4}$ gick det att köra. Men resultatet vid denna körning är inte intressant eftersom den inte fick träna tillräckligt. Resultatet blev i att 99% av alla datapunkter hamnade i ett kluster.

A.5 Test B5

Metod: *Xmeans* **Parameter:**10-40 **Avståndsmätning:***Euklidskt* **Datamängd:***B*



Figur A.5. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

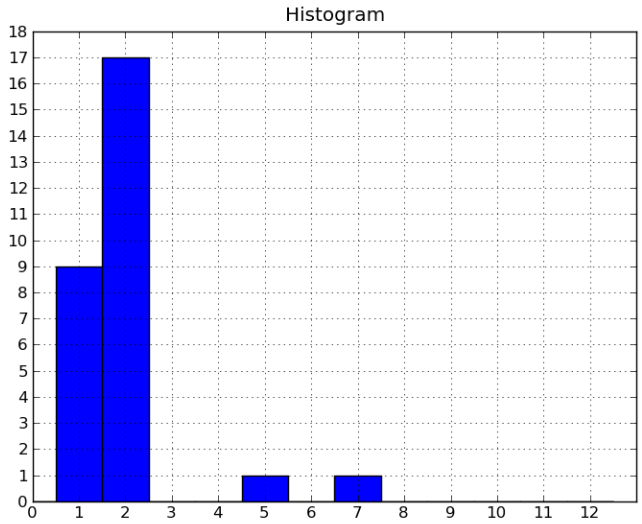
EVAL12-Medelvärde ≈ 2.35

B C E G H K O Q S X Z	A J
C K U V X Y	B C D E G K O Q R S X
F P T	B D H K L N R S X Z
C E L Z	D O P R
H M N U	H M N
B D E K R S Z	I J L
B G K R	B G Q S X Z
M W	N U V W
I T V Y	F N T U V W Y

Tabell A.4. Bokstäver som har dömts lika.

A.6 Test B6

Metod: *Xmeans* **Parameter:**24-26 **Avståndsmätning:***Euklidskt* **Datamängd:***B*



Figur A.6. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

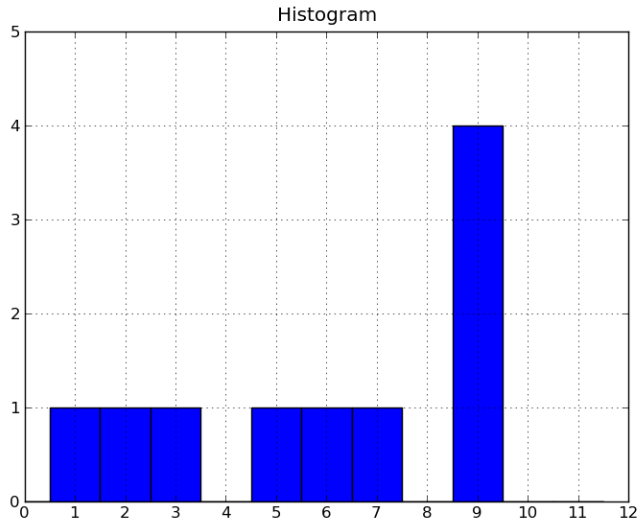
EVAL12-Medelvärde ≈ 1.964

T V Y	F T V Y
B D G O Q R	I J
T V W Y	F P
B R	N V W
C E	S X Z
G Q	B D H K R S Z
C U	M W
A J	B G K O Q R X
I J L	M N W
K U X	B C E G O Q X Z
B D R S Z	F P
H M N U	

Tabell A.5. Bokstäver som har dömts lika.

A.7 Test S1

Metod: *K-means* **Parameter:**10 **Avståndsmätning:***Euklidskt* **Datamängd:***S*



Figur A.7. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

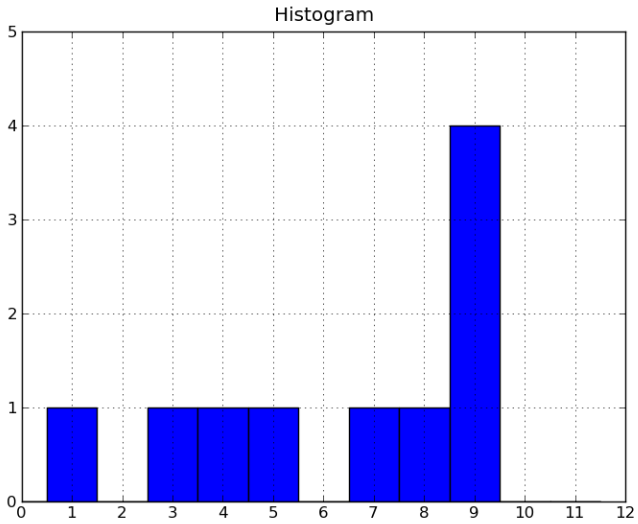
EVAL12-Medelvärde ≈ 6.0

1 8
3 5 8 9
1 5 8
7 9

Tabell A.6. Bokstäver som har dömts lika.

A.8 Test S2

Metod: *K-means* **Parameter:**10 **Avståndsmätning:***Cosinusmått* **Datamängd:***S*



Figur A.8. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

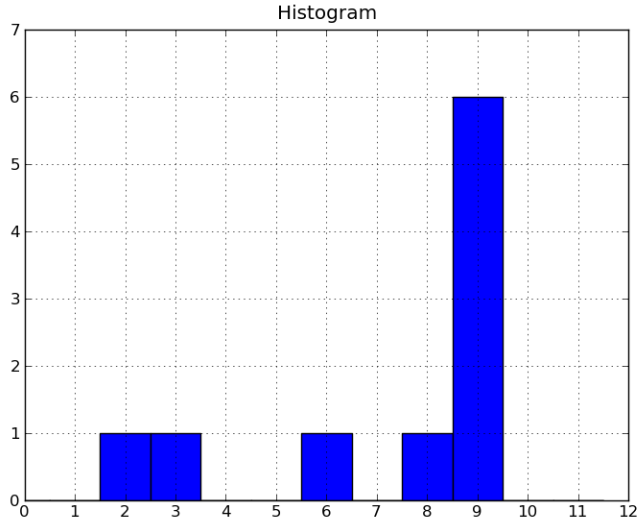
EVAL12-Medelvärde ≈ 6.4

1 2
3 5 8 9
1 9
1 8

Tabell A.7. Bokstäver som har dömts lika.

A.9 Test S3

Metod: *K-means* **Parameter:**10 **Avståndsmätning:***Manhattan* **Datamängd:***S*



Figur A.9. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

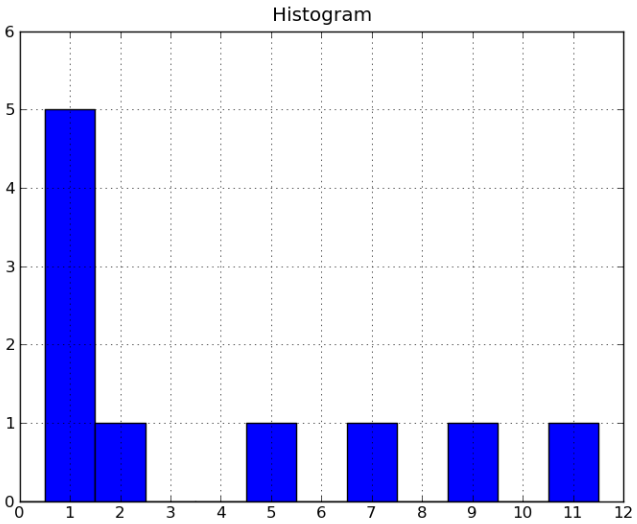
EVAL12-Medelvärde ≈ 7.3

1 9
3 5 9
1 2
1 8

Tabell A.8. Bokstäver som har dömts lika.

A.10 Test S4

Metod: *Hiearkisk* **Parameter:**10 **Avståndsmätning:***Euklidskt* **Datamängd:***S*



Figur A.10. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

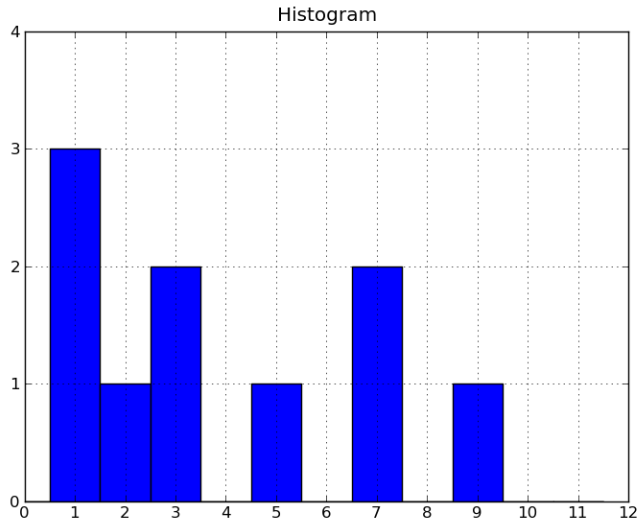
EVAL12-Medelvärde ≈ 4.0

1 3 9
1 2 3 9
3 5 9
1 4 8
7 8
2 5 8

Tabell A.9. Bokstäver som har dömts lika.

A.11 Test S5

Metod: *Hiearkisk* **Parameter:**10 **Avståndsmätning:***Manhattan* **Datamängd:***S*



Figur A.11. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

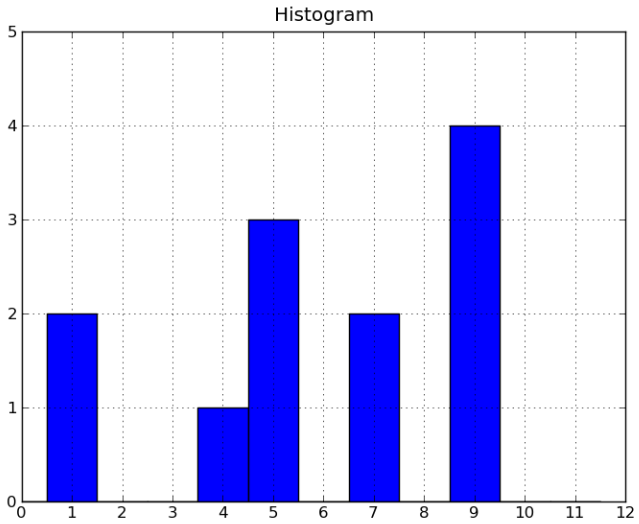
EVAL12-Medelvärde ≈ 3.9

5 9
5 7
1 4 9
2 3 9
1 2 5 8
1 4

Tabell A.10. Bokstäver som har dömts lika.

A.12 Test S6

Metod: *Xmeans* **Parameter:**6-12 **Avståndsmätning:***Euklidiskt* **Datamängd:***S*



Figur A.12. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

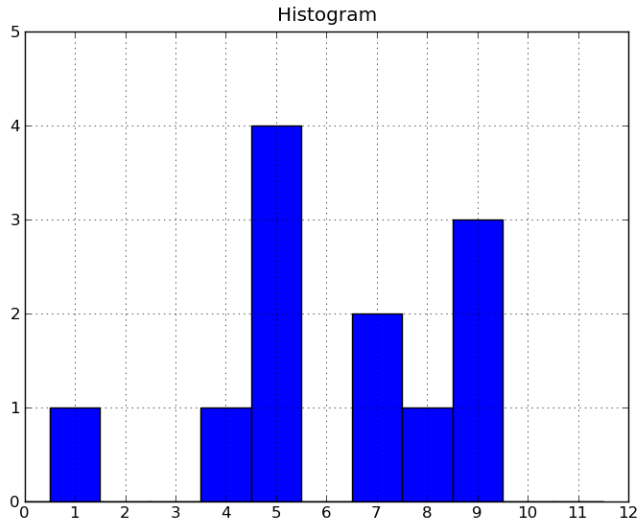
EVAL12-Medelvärde ≈ 5.916

5 8 9
1 8
1 9

Tabell A.11. Bokstäver som har dömts lika.

A.13 Test S7

Metod: *Xmeans* **Parameter:**6-12 **Avståndsmätning:***Manhattan* **Datamängd:***S*



Figur A.13. Detta histogram visar antalet kluster som har fått respektive EVAL12 värde.

EVAL12-Medelvärde ≈ 6.166

5 8 9
1 8
3 9

Tabell A.12. Bokstäver som har dömts lika.

Bilaga B

Formler

Skalärprodukt Skalärprodukt är definierat som :

$$a \cdot b = |a||b|cos\alpha = \sum_{i=1}^{n}(a_i b_i) = a_1 b_1 + a_2 b_2 + ... + a_n b_n \tag{B.1}$$

I denna undersökning är det intressant att lösa ut $cos(\alpha)$;

$$cos(\alpha) = \frac{a \cdot b}{|a||b|} \tag{B.2}$$

Längd av en vektor Notation för längd av en vektor är $length(a) = |a|$
Längden av en vektor är definierat som :

$$|a| = \sqrt{a_1^2 + a_2^2 + ... + a_n^2} \tag{B.3}$$

Normalisering av vektor Normalisering innebär att man skalar om vektorn så vektors längd blir 1.

$$norm(a) = (\frac{a_1}{|a|}, \frac{a_2}{|a|}, ..., \frac{a_n}{|a|}) \tag{B.4}$$

Trigometrisk identitet

$$\begin{aligned} sin^2(x) + cos^2(x) &= 1 \\ sin(x) &= \pm\sqrt{1 - cos^2(x)} \end{aligned} \tag{B.5}$$

Trigometrisk identitet

$$cos^2\left(\frac{\theta}{2}\right) = \frac{1 + cos(\theta)}{2} \tag{B.6}$$

Litteraturförteckning

[1] Wikipedia,*Övergripande information om klustringsalgoritmer*
http://en.wikipedia.org/wiki/Clustering_algorithm
(Juni 2011)

[2] Wikipedia,*Evaluation of clustering*
http://en.wikipedia.org/wiki/Cluster_analysis#Evaluation_of_clustering
(Juni 2011)

[3] J. MacQueen, *Some methods for classification and analysis of multivariate observations*
http://projecteuclid.org/DPubS/Repository/1.0/Disseminate?view=body&id=pdf_1&handle=euclid.bsmsp/1200512992
(Juni 2011)

[4] Kiri Wagstaff,Claire Cardie(2001 *Constrained K-means Clustering with Background Knowledge*
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.4624&rep=rep1&type=pdf
(April 2011)

[5] UCI,*UCI machine learning repository*
http://archive.ics.uci.edu/
(April 2011)

[6] David J. Slate, *Datamängden som kommer användas*
http://archive.ics.uci.edu/ml/datasets/Letter+Recognition
(April 2011)

[7] E. Alpaydin, C. Kaynak, *Datamängd som kommer användas*
http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits
(Juni 2011)

[8] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); *The WEKA Data Mining Software: An Update; SIGKDD*

Explorations, Volume 11, Issue 1.
`http://www.kdd.org/explorations/issues/11-1-2009-07/p2V11n1.pdf`
(April 2011)

[9] Clifton, Christopher (2010),*Encyclopedia Britannica: Definition of Data Mining*
`http://www.britannica.com/EBchecked/topic/1056150/data-mining`
(April 2011)

[10] Peter W. Frey and David J. Slate(1991),*Letter recognition using Holland-style adaptive classifiers*
`http://www.springerlink.com/content/x83328826p16u32u/fulltext.pdf`
(April 2011)

[11] Wikipedia,*En beskrining av bitmaps*
`http://sv.wikipedia.org/wiki/Bitmap`
(april 2011)

[12] Harri Valpola(2000-10-31),*Supervised vs. unsupervised learning*
`http://users.ics.tkk.fi/harri/thesis/valpola_thesis/node34.html`(April 2011)