Det Teknisk-Naturvidenskabelige Fakultet

Aalborg Universitet

Basisuddannelsen

TITEL:

Spam skal bekæmpes!

Projektperiode:

P1,
7. Oktober - 16. December, 2002

Projektgruppe:

A325

Gruppemedlemmer:

Arne Mejholm

Bjarne Mark Kondrup

Dan Malthesen

Lars Ole Christensen

Jens Peter Bak

VEJLEDERE:

Jens Dalgaard Nielsen
Palle Qvist

Synopsis:

Formålet med projektet er at give større indblik i fænomenet spam, og de nuværende løsningsmodeller. Rapporten giver indblik i generel bayesiansk teori. Yderligere har vi udviklet vores egen løsningsmodel, hvilket tager form som et spamfilter, inplementeret i "SML". Det er forsøgt implementeret udfra et naivt bayesiansk netværk.

Antal kopier: 13

Rapport – sideantal: 72

Appendiks – sideantal: 7

Totalt sideantal: 79

Spam skal bekæmpes

A325

16. december 2002

Kapitel 1

Forord

I forbindelse med udarbejdelsen af denne rapport vil vi gerne takke, vores hovedvejleder Jens Dalsgaard, som har været en gode støttet for vores gruppe når vi havde tekniske sprøgsmål.
Til Palle Quist vores bivejleder, som har hjulpet os på med at få styr på strukturen gennem hele forløbet og med de kontekstuelle sprøgsmål vi måtte have.
Til Wladyslaw for et rigtig godt interview som vi havde med ham torsdag d.22/11.
I forbindelse med rapporten er der udarbejdet et program der kan filtrere emails, den nyeste udgave kan findes på:

`www.but.auc.dk/~arme0233/SMLBAYES/smlnaivebayes.tar.gz`

God læselyst...
Mvh A325

Indhold

Kapitel 2

Indledning

Den teknologiske tidsalder vi lever i er fyldt med kreative kræfter, der trækker i alle retninger. Normer og etik er nogle mærkelige størrelser, hvis man befinder sig på et nyt territorium. Internettet er et relativt nyt medie, normerne og etiken er kun ved at tage deres spæde form. Hvis man som fordomsfrit individ, langt væk fra elektroniske hjælpemidler og inden for lovens rammer, lapper til sig af livets sødme, uden tanke for konsekvenserne af ens handlinger, får man tit slag. Tænder man for sin computer og som fordomsfri "surfer" på Nettet, tilmelder sig de steder man har lyst til, bliver man bombarderet med elektroniske reklamer, fra steder man ikke har bedt om. Det er den barske realitet ultimo 2002. Hvis man som erfaren Internet bruger, ville være i stand til at være fri for spam, ville spam problemet måske være tålelig, men situationen er mere kompliceret. Deltager man i nyhedsgrupper eller blot har sin egen hjemmeside på Nettet, er man udsat for spammails. De teknologiske kræfter trækker i en negativ retning, når blikket falder på e-mail. Der er en lovgivning på området, men den er svær at håndhæve, eftersom spammeren ofte sidder i udlandet Hvordan går det til, at vores mail server skal belemres med e-mails fra folk vi ikke kender, eller vil have noget med at gøre? Svaret på dette er ganske ligetil: der er penge i spam, mange penge. Den optimale computer ville være i stand til fejlfrit, at frafiltre den e-mail man ikke ønsker og lade de lødige slippe igennem. Det er utopi at forvente det bliver en realitet, der er et marked for spam, med de mange forretningsfolk der vil slå mønt på folk. Teknologibærerne bag spam, vil hele tiden forsøge at være på forkant med de præventive midler, der bliver sat ind for at stoppe spam.

Vores gruppe har arbejdet indædt for, at granske hele spektraet inden for problemstillingen omkring spam. Vi har prøvet at belyse de samfundsrelaterede problemer, der er forbundet med fænomenet spam. Vores oplagte intensioner var, at undersøge markedet for eksisterende spam bekæmpende resourcer, hvorefter vores nyvundne viden, har sat os i stand til at bidrage med et løsningsforslag, der er udformet siddestillende med de kundskaber vi har lært undervejs. Vi havde som målsætning intensioner om at lave en generel skelet til et program, der kan fra filtrere spammails.

Vores udgangspunkt starter med at belyse generelle reklame former.

Kapitel 3

Reklame former

Spam: En uanmodet automatiseret reklame e-mail. For at tydeligøre problemstillingen omkring spam vil vi først indledende tage udgangspunkt i de medier man konventionelt finder reklamer på skrift. Traditionelt finder man reklamer i aviser, ugeblade, fagblade og omdelte reklamekataloger i form af trykte billeder eller tekst. I tv finder man reklamer som billedsekvenser der, alt efter kanalens politik, afbryder det planlagte program. Med udbredelsen af Internettet er der kommet adskillige nye måder at reklamere på. Meget anvendt er, at reklamer betaler for at få en animeret rubrik på en hjemmeside. Denne metode findes også i en udgave, når man åbner en hjemmeside, samtidigt starter en reklame hjemmeside også kaldet en "pop up ad". Fælles for disse er, at modtageren på et eller andet tidspunkt selv accepterer, at modtage disse reklamer, hvad enten det er i form af et abonnement på et tidskrift, eller når vedkommende tænder for tv'et. Når modtageren søger på internettet er vedkommende også aktiv i den henseende at vedkommende åbner en hjemmeside på producentens vilkår.

Med internettet er der også kommet en ny reklameform, nemlig reklame via e-mail, hvilket i sig selv ikke er noget dårligt fænomen. Man kan tilmelde sig et firmas mailing liste og få tilsendt gode tilbud, hvis man er interesseret i det. Desværre kan dette også misbruges og det er her spam kommer ind i billedet.

Spam er essentielt uanmodet e-mail reklame som bliver sendt uden nogen form for samtykke. Folk bliver oftest ofre for spam reklame, fordi de har ladet sig opskrive på et pseudo firmas mailing liste, altså et firma, der hvis eneste hensigt med mailing listen er at opfange så mange e-mail adresser som muligt. Denne liste har ofte en meget omfattende størrelse, hvilket gør den til en potentiel handelsvare. De personer, der ønsker at udsende spam køber en såkaldt "spam-mailing-liste", hvorved de får et vist antal e-mail adresser. Spammerne udsender så uden hensyn til alder, køn eller anden betænkning, reklame for deres firma. Fordelen ved spam e-mail er at hvor traditionel reklame har store udgifter forbundet hos senderen, i form at afgifter, papir udgifter og arbejde med tilblivelsen at reklamen [9]. På nær den sidste er disse udgifter ikke gældende for spam. De få udgifter der er gældende for spam reklame, er typisk knyttet til Internet abonnement og den tid det tager at sende et givent antal mails. Denne udgift, føler den almene bruger nok ikke, men for en ISP [1] er der langt større udgifter forbundet med den ekstra trafik, der går gennem deres servere. Problemstillingen ved spam vil dog blive diskuteret nærmere i et følgende kapitel.

3.1 Historien bag spam

Udtrykket spam kommer fra den amerikanske fabrik Hormel Foods som fremstiller skinke på dåse den såkaldte *Shoulder Pork and ham* eller *Spiced Ham* den er siden blevet adopteret som et meget beskrivende termonologi inden for Internet/computer-verdenen. Spam udtrykket defineres ubevidst i en sketch lavet af de engelske komiker fra Monty Python's Flying Circus, hvor en det meste af sketchen går ud på at få sagt spam så mange gange som muligt, og give et billede af en restaurant, hvor der kun bliver lavet retter med spam(dåse skinke). Hvis man kigger på følgende linje som er en del af sangen fra sketchen får man straks en association til nutidens *uønskede automatiserede reklame E-mails*:

> "Spam, spam, spam, spam, spam, spam, spam, spam, lovely spam! Wonderful spam!"

Denne sang blev grundlaget for at man idag forbinder spam med noget som gentager sig mange gange og er yderst irriterende.
MUD[2] brugere blev dem der navngav de automatiserede meddeleser som spam. MUD om er en udvikling af RPG [3] som Dungeons and Dragons. Spam udtrykket blev en populær betegnelse til *bots* som automatisk sendte store mængder af meddelser i chatprogrammet. Men allerede i 1978 blev den første e-mail spam sendt ud af Einar Stefferud. Mailen var en invitation til receptioner i Californien, på baggrund af at hans firma var kommet med en ny maskine DEC-20. Denne meddelelse blev sendt til alle ARPANET brugere på USA's vestkyst, Einar Stefferud blev natuligvis ikke beteget som spammer på daværende tidspunkt.

`jj@cup.portal.com says: HELP ME!`

En kedelig opfindelse kom den 24. maj 1988 da Rob Noha, som den første på USENET masse-udsendte en post, der beskrev hvordan hans *College Fund* var løbet tør for penge og dermed bedte om at folk ville sende penge for at hjælpe ham. Det var forløberen for hvad vi idag, kender som bondefangeri på nettet.
Make Money Fast, dette er en af de spams, der hurtigt blev stærkt udbredt i starten af 90erne, en spam som blev bredt ud over USENET ved hjælp af kædebrevs strategien. Den 18 jan 1994, stod en student for USENETs første rigtige spamangreb, hvor hver eneste Newsgroup blev spammet med meddelsen:

[1] Internet Service Provider
[2] Multi User Dungeons
[3] Role playing games

`Global Alert for All, Jesus is Coming Soon.`

Ca. 4 måneder senere kom der en meget omtalt spam, som omhandlede opholdstilladeler, hvor igen hver eneste nyhedsgruppe i USENET blev offer for endnu et spam angreb. Denne situation var anderledes fordi de to ophavsmænd ikke stak halen mellem benene, men derimod var stolte af deres spam. Dette resulterede i bogen: “How to make a fortune on the information superhighway” med undertitlen “Everyones Guerrilla Guide to Marketing on the Internet and Other Online Services”. Heldigvis blev bogen en fiasko og ophavsmændene forsvandt ud af verdensbilledet.
Spammen der dengang startede, har idag ødelagt folks tiltro til nyhedsgrupper, der er ikke mere nogen som opgiver deres rigtige e-mail adresser, dem der gør ender som regel med en bunke af uønsket mail. For at gemme sig for automatiske programmer der samler e-mail adresser og tilføjer dem til store spam-lister, er det nu almindeligt at tilføje fx. NOSPAM til sin e-mail adresse (borgeNOSPAM@hotmail.com). Dette burde ikke være nødvendigt, men ser ud som om spam er kommet for at blive! Alt information er baseret udfra [19]

Kapitel 4

E-mailens historie

[23] I 1960'erne savnede det Amerikanske militære forsvar en netværksforbindelse til de forskellige institutioner, som de samarbejdede med,da de var i krig med Sovjetunionen. De forskellige institutioner havde deres eget netværk uden fælles kommunikation. Man udviklede netværket, hvor der skulle være en fælles kommunikation. Den oprindelig ide var at man skulle få et netværk der kunne overleve et atom angreb, dvs. at hvis en enkelt del af internettet faldt ud, skulle resten af verden stadig kunne kommunikere.

Det første netværk blev kaldt ARPANET [1], Network Control Protocol.ARPANET forsvandt helt i 1990 da ideen blev udviklet med kommunikations protokollen TCP/IP, hvor det i dag er muligt at kommunikere med hele verden. Da IP adressen kun bestod af numre var det svært at huskede til tider meget komplicerede adresser, brugte man informations systemet www. Brugerne har adgang til forskellige filer på www siderne, hvor man enten kan hente dem ned på sin egen computer.

Applikationer og tjenester på Internettet er meget hurtige og på nogle punkter udvikles det til misbrug, fx. hacking, spam osv.

E-mail var den første kommunikationsform, der var tilgængelig for brugerne på Internettet. Man udviklede kommunikationen med e-mails, til diskussioner og chat forum. E-mail er dog stadig en meget benyttet kommunikations form.

I april 1994 blev der sendt den første spam mail rundt af advokaterne Laurence Canter og Martha Siegel fra Arizona. De reklamerede for juridisk bistand til deltagelse i det amerikanske opholdstilladelseslotteri til diskussionsgrupper på usenet news. Senere udviklede det sig til at man tog nogle e-mail adresser fra forskellige brugere på usenet og sendte uopfordret reklamer rundt. Internettet er blevet mere og mere kommercielt pga.af spam.

[1] The Advanced Research Projects Agency

4.0.1 Vigtige Årstal:

- 1969: De første 4 servere bliver opsat ved forskellige universiteter og forskningsinstutter rundt om i USA.
- 1971: ARPANET forøges til femten servere inklusiv NASA.
- 1972: Ny kommunikationsform Email og antallet af servere er steget til 32 stk.
- 1973: ARPANET etableres i England og Norge.
- 1982: Transmission Control Protocol (TCP)og Internet Protocol (IP) oprettes til brug på ARPANET.
- 1983: ARPANET deles op til ARPANET og MILNET (det militære net)
- 1984: Antallet af servere overskrider 1000.
- 1986: NSFNET opstår med fem supercomputere som knudepunkter. De tilbyder højhastigheds datatransmission.
- 1989: Antallet af servere overskrider 100000.
- 1990: ARPANET nedlægges.
- 1991: CERN frigiver World Wide Web, antallet af server overskrider 500000.
- 1993: Mosiac, den første grafiske webbrowser, bliver lanceret. Dette medfører, at der kommer en stigning af web-trafikken. UNI-C den første danske Internet udbyder, hvor almindelige mennesker fik adgang til Internettet.
- 1994: Online indkøb kommer frem og browseren Netscape lanceres.
- 1995: 5.000.000 servere på Internettet.
- 1997: Realvideo lanceres, hvor der kan ses video over nettet, antallet af servere overkrider 10 milioner.
- 1998: Antallet af brugere overstiger 150 millioner.

Årstal set på [23]

4.1 E-Mails

4.1.1 Opbygning af en e-mail

Hovedkilde er [18]
Hvordan er en e-mail opbygget ? En e-mail består af 2 dele. "Header" og "Body". Headeren indeholder informationer, der er vigtig for e-mailens forsendelse og levering. Herunder er f.eks. den dato den er sendt, afsenderens e-mail adresse, modtagerens e-mail adresse og overskriften på beskeden. Yderligere header information tilføjes i mange tilfælde af ens e-mail program, som, der i mange tilfælde tilføjer et Id. nummer til beskeden, og tit information omkring hvilket program den blev skrevet i. Sådanne ekstra headers tilføjes også af de forskellige mailservere som e-mailen kommer igennem på sin vej. Eksempler på e-mail klienter, er f.eks. Microsoft Outlook (og Outlook express), Eudora Pro, Netscape Messenger, Pine, elm, mutt og mail. Sidst nævnte fire er alle fortrinsvis brugt på "Unix" systemer, men findes ligeledes til andre operativsystemer. Et eksempel på en e-mails opbygning kunne se således ud, dog kan nogle felter (specielt i headeren) skrives på flere former som alle accepteres.

```
Date: sun, 22 November 2002 13:49:25 +0100
From: mig@potteplante.com
To: Vejleder <vejleder@skolen.com>
Subject: Blot en lille test. Ikke noget nyt...

Hej Vejleder!
Vi ville blot lige sende en lille test e-mail.
Dette er ikke spam ;)

Med Venlig Hilsen
Gruppe A325
```

4.1.2 Hvordan sender man så e-mails?

Efter man har skrevet sin e-mail, som for det meste sker ved hjælp fra sit e-mail klient program (se tidligere eksempler), trykker man på ikonet for send. Nu vil programmet først give beskeden til ens lokale mailserver. I de fleste tilfælde ligger denne hos ens egen Internet udbyderen, eller lokalt i f.eks. ens firma. Endvidere kan man have en privat mailserver hjemme. Der er mange muligheder... Denne mail server modtager så e-mailen og behandler den ved at forsøge at sende til modtageren. Begge servere "snakker" sammen via en protokol. Disse protokoller kan ses som sproget og formen for hvordan samtalen mellem maskinerne finder sted.

Nogle af de almene protokoller er her:

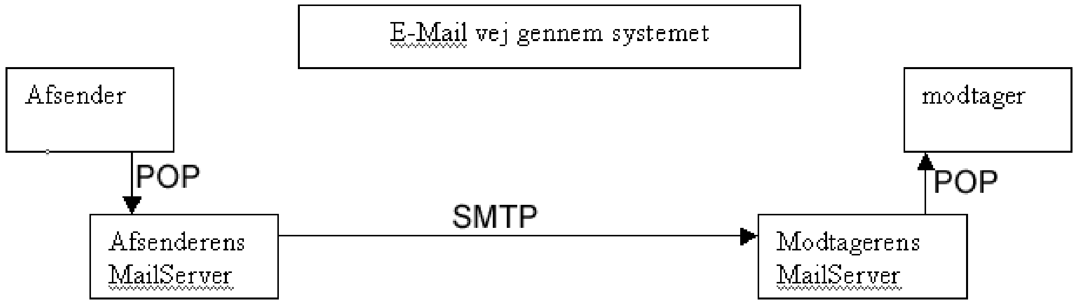
- SMTP - Simple Mail Transfer Protocol - Denne protokol bruges hovedsageligt til kommunikation mellem serverne.
- POP - Post Office Protocol - Denne protokol bruges hovedsageligt til kommunikationen mellem klienten og mailserveren.
- IMAP - Internet Message Acces Protocol - Også protokol til afsendelse af e-mails. Et program der kan "snakke" IMAP kan manipulere postkasserne på serveren.
- NNTP - Network News Transfer Protocol - Denne protokol er brugt til "Nyhedsgrupper". Usenet news servere bruger den til afsendelse af nyheder mellem hinanden, og news readers (klient programmet) bruger denne til at hente og sende news til serverne.

Når man skriver sin e-mail på computeren, sker dette i et klientprogram. Denne kaldes også for et MUA [2]. Den sender som sagt mailen videre til serveren, som fungerer som en slags postkontor, der sørger for videre afsendelse.
En Mail Server,også kaldt MTA [3] "snakker" så sammen med modtagerens MTA og sørger for at videregive sine mails.En sådan samtale mellem 2 servere er ikke så forskellig fra en samtale mellem 2 mennesker. Den foregår i nogle trin, som kan forsimples til nedenstående. Bemærk at der ikke er benyttet nogen specifik protokol til nedenstående eksempel det er blot en skitse over en standard "samtale" uden noget ekstra eller specifikt for nogen protokol.

1. Introduktion, De 2 servere præsenterer sig for hinanden
2. Afsenderens mailserver identificerer afsenderen
3. Afsenderens mailserver identificerer modtageren
4. Modtagerens mailserver prøver at finde modtageren, hvis ikke afbrydes forbindelsen
5. Afsenderens mailserver sender "data" (mails) til Modtagerens mailserver
6. De siger farvel

Når de siger farvel, har afsenderen "afleveret" posten til modtagerens e-mail server, som så sørger for at videregive posten til den pågældende bruger, som den er adresseret til.
Selve afsendelsen af e-mail fra start til slut, kan ses på nedenståene billede 4.1

[2] Mail User Agent
[3] Mail Transport Agent



Figur 4.1: En E-Mails vej gennem systemet.

4.1.3 Andre headers

Der findes andre headers udover de mest basale, og nødvendige som vi indtil nu har koncentreret os om.

- Cc og Bcc - Carbon Copy og Blind Carbon Copy. Bcc tillader at sende samme mail til flere brugere samtidig, uden at modtagerne af mailen kan se hvem den ellers er sendt til. CC tillader at modtageren kan se hvem der ellers har fået den samme mail...
- Reply-To - Denne indeholder en adresse hvortil svaret skal sendes, såfremt det ikke skal returneres til den originale afsender. Eksempel: Hjem@Hjem.dk sender til Skole@skole.com og ønsker svaret sendt til Arbejde@arbejde.com
- Sender - Denne bruges til f.eks. en mailingliste. Hvor From ændres til at være personen som ønskede mailen afsendt, og sender indeholder selve mailing listens adresse.
- X- headers - X-headers er ikke-standard headers, som brugerne selv kan opfinde. Der er ingen risiko for at MTA misfortolker en sådan header, da ingen standard headers begynder med X-.

4.1.4 Hvordan hentes e-mails

Når man skal hente e-mails fra sin mailserver, foregår det på stort set samme metode som, når serverne sender til hinanden. Proceduren er således mellem "bruger" og "server". Igen skal det bemærkes at ingen specifik protokol er benyttet og at eksemplet blot er en hypotetisk "samtale".

1. bruger identificerer sig overfor server
   (a) 1.1 Bruger sender username

(b) 1.2 Bruger sender password

2. Server accepterer (hvis dette er tilfældet) bruger, og giver denne adgang til videre kommunikation. Ellers afbrydes kommunikationen her

3. Bruger spørger hvad der ligger på server.

4. Mail Server svarer med antal mails og størrelse der ligger klar. Sender evt. også headers og andet anmodet information.

5. Bruger beder om at hente mails (enkelte eller alle)

6. Server sender efter anmodning mails

7. Bruger beder efter modtagelse af mails server om at slette serverens kopi

8. Bruger logger af

Nogle af punkterne kan omgås, og tilpasses efter behov, alt afhængigt af programmet. F.eks. behøver man ikke at slette serverens kopi af e-mailen, men kan i stedet gemme en "backup" på serveren. Man kunne også bare få antallet af e-mails at vide og ikke hente nogen, eller kun hente nogle udvalgte og slette andre. Mange anti-spam programmer bruger den metoden at de logger på serveren, og ser hvilke mails der ligger der. De henter nu ikke alt, men kun f.eks. 20 linier af hver e-mail, hvorved trafikken bliver begrænset. Disse 20 linier pr. mail bliver så analyseret sammen med deres header for tegn på spam, og brugeren får vist resultatet. Brugeren kan nu uden at skulle hente de resterende dele af spam beskederne, slette dem fra serveren, og derved spare plads og båndbredde. Yderligere kan mange virus, og skadefulde programmer herved undgåes, da der relativt set ikke "hentes" noget ned til computeren, og da vedhæftede filer ignoreres

Kapitel 5

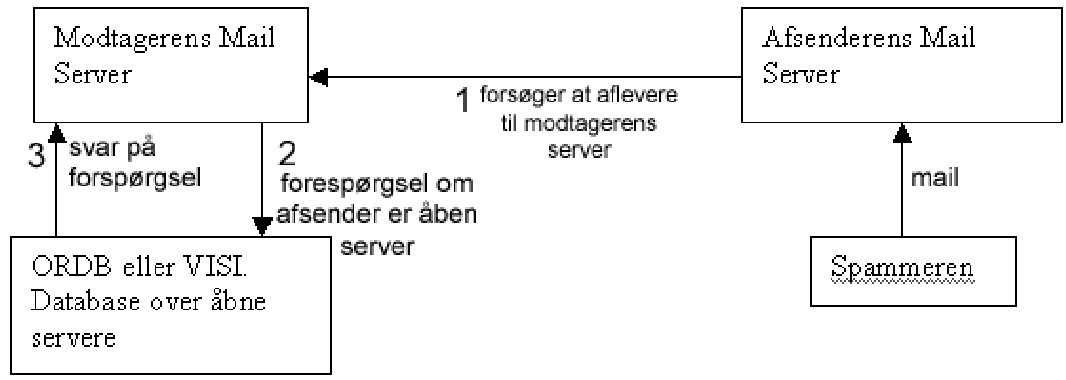
Spammers

5.1 Teknologibærer

Teknologien bag masseudsendelsen af e-mail kan opdeles i to hovedafdelinger. (1) programmer der gør det muligt. (2) måderne hvorpå aktøren kan slippe af sted med det. Den danske It-virksomhed Macroware [25], har lavet en e-mail robot, der relativt hurtigt kan finde og lagre e-mail adresser og selv sende e-mails. Programmet er brugbart på søgemaskinen Altavista. Man kan få programmet til at søge efter bestemte ord, f.eks. bomuld, der vil så dukke ca. 5900 hjemmesider op, programmet kan herefter gennemgå disse hjemmesider for e-mail adresser. Finder programmet en e-mail adresse, kan det lagre den og sende en mail til den. Dette program er trukket tilbage fra markedet pr.220502, efter Macroware selv blevet bombarderet med spam. Macroware var blot en, ud af adskellige spam software udbydere, men er blevet nævnt fordi den var dansk.

Måderne hvorpå spammernes anonymitet bliver bevaret:

Det er let og ikke forbundet med personlige spørgsmål at oprette en Hotmail adresse, det er herefter uden risiko, for personligt at blive gjort ansvarlig, at man kan udsende sine mails til de modtagere man ønsker.2.Åbne servere er et stort problem. Computere på nettet der ikke selv er sat op til, eller er i stand til at levere elektronisk post direkte til modtageren [20], kan sende posten til et mail relay, som så finder ud af, at sende posten videre. En mail server er overordnet enten opsat til ukritisk at sende alle mails videre i systemet, det kaldes en åben server, eller kun til at sende mails videre, hvis afsenderens IP adresse er registreret i serveren, det kaldes en lukket server. Spammere benytter i stor stil disse åbne relays, til at få deres spam udsendt. Spammerne kan her uhindret sende deres mails totalt anonymt, og tillige, hvis spammeren ønsker det, i et intetanende firmas navn. Spammerne kan skjule deres identitet når de sender deres mails, ved at bruge talrige åbne servere i kombinationer med hinanden. Der er som før beskrevet mange servere, der ubevidst er opsat som en åben server.



Figur 5.1: Procuduren for open relay checking

Der er forskellige organisationer, firmaer og enkeltpersoner, der laver optegnelser over hvilke mailservere der er åbne. ORDB [1] og VISI er eksempler herpå. Mange af disse optegnelser er frit tilgængeligt på nettet. Problemet med dette åbenlyse gode initiativ, er blot at spammerne også kan indhente disse oplysninger og bruge dem effektivt.

Proceduren for registrering af åbne mailservere er at scanne serverne. Servere er blevet scannet af ORDB, for at finde ud af om de er åbne. Det er ifølge Wladyslaw [15] problematisk eftersom det er forbudt i mange lande. ORDB har ifølge Wladyslaw scannet port 25 ved at sende nogle falske E- mails, og derefter tjekket de udvalgte serveres modtagelighed.. Det kan være svært at scanne de forskellige servere og entydigt bestemme om hvorvidt de forskellige servere er åbne eller lukkede, eftersom de har egen konfiguration. Et godt eksempel på ovenstående problem, er MIT [2], som er et stort Universitet. De har ifølge Wladyslaw bestemt ikke en åben server, men blev af ORDB betragtet som om de havde en åben server, fordi deres opsætning var meget speciel og derfor svært gennemskueligt. Det resulterede i en retsag mellem ORDB og MIT. MIT tabte og blev nødt til at skifte ISP. Det er blot et bevis på at det er svært at afgøre hvem, der skal bestemme om, hvorvidt de forskellige servere er åbne eller ej. Det er et spørgsmål om tillid, moralske skrupler og ansvarlighed. Aalborg Universitet abonnerer på RBL [3] plus listen, som er en åben server blackliste, Universitetet har den aftale med plus listen at hvis, der kommer klager over nogle afvisninger er det RBL udbyderen, der har det juridiske ansvar, de får penge for det. Udover eksisterende virksomheder som den danske It-virksomhed Macroware,der producerer og sælger softwaren til masseudsendelser af E-mail, er der også virksomheder der mod en begrænset betaling tilbyder folk, der har interesse i at få udsendt relativt store

[1] Open Relay Databse
[2] Massachusetts Institute of Technology
[3] Realtime Blackhole List

mængder post, denne service, mod en begrænset betaling. Software virksomheden E-mail Marketing er en sådan udbyder. [22] Deres produkt hedder ghostsender. Ghostsender kan sende relativt store mængder E-mails ud på en beskeden tid, hvis man ønsker det, men man kan også sende moderate mængder inden for en uangivet tidsramme. De lover deres potentielle kunder at hvis anonymitet en nødvendighed, er det ikke et problem. De opererer med overkommelige priser, som det kan ses i det nedenstående:

Total	E-mails Credits	$ per million E-mails sent
$99	500,000 E-mails	$198.00 per million
$149	1 Million E-mails	$148.00 per million
$199	2 Million E-mails	$99.50 per million
$399	5 Million E-mails	$79.80 per million
$649	10 Million E-mails	$64.90

E-mail Marketing påpeger over for deres kunder, at det ikke er tilladt at sende spam, de opfordrer også deres kunder til ikke at gøre det, men skulle kunden gøre det alligevel, er det kundens eget ansvar. Man kan let grine af deres dobbelt moral, for samtidigt med deres flotte opfordringer, byder de sig selv til at gøre det beskidte arbejde. Man skal blot indsende ens filer med de e-mail adresser man ønsker skal modtage ens reklame. Hvis man vil have udarbejdet en målrettet e-mail liste kan de også få lavet en sådan, eller man kan købe 100 millioner e-mail adresser og programmet til at udsende ens reklame til dem for 1000$.

5.2 Spam aktørerne

I følge artikel [11] anser mange spammere Internettet som en offentlig resource, uden diktatoriske spilleregler.
De amerikanske skatteydere har indirekte betalt for den research, der har frembragt Internettet. Det amerikanske militær havde på et tidspunkt brug for en alternativ kommunikationsmulighed, der kunne fungere på samme høje funktionalitets niveau som telefonen. dette behov førte til opfindelsen af de protokoller, 4 som Internettet bygger på. Denne økonomiske kendsgerning er en vigtig og relevant grund til de amerikanske spammeres opfattelse af Internettet som en offentlig resource. Spammerne mener at befolkningen skal opmuntrer dem til at fortsætte deres masseudsendelser af reklame e-mail. Grunde hertil er følgende:

- Denne form for post er mere skånsomt overfor miljøet og mindre nedslidende for postomdelerne, end den traditionelle personlige uddeling er.
- Der bliver ikke brugt papir til at trykke reklamerne.
- Den traditionelle reklame omdelings enorme brændstofs forbrug bliver sparet.

- De menneskelige resurser til at bringe reklamerne ud bliver sparet.
- Det er lettere at målrette posten.
- Det er mindre opslidende at slette en e-mail end at smide ens brevsprække reklamer væk.

Spammere føler sig tit retfærdiggjort ved kun at sende et par hundrede e-mails til bestemte udvalgte, de har jo begrænset sig. En spammers succes kriterier er svære at acceptere, hvis man som internetopkoblet føler sig generet af spam, men de er ligetil og lette at forstå. Hvis køberesponsen som et minimum er ca. 1 promille af de udsendte reklamer, er det en succes. Hvis den udsendte reklame modtages af en internetopkoblet, der ikke aktivt har gjort spammeren opmærksom på sin modvilje over modtagelse, betegnes det som en succes, for det kan være en pottentiel kunde.

5.3 Hvem bemidler Teknologibærerne

Problemet med spam er at der er penge i skidtet, der er mange firmaer, der er villige til at betale for at få udsendt deres reklamer, der er ganske enkelt tale om udbud og efterspørgsel. Efterspørgelsen er til stede ved firmaernes betaling for en udsendelse, og når der er nogle, der er villig til at betale for en ydelse, er der altid nogle, der er villig til at tage pengene og udføre arbejdet. Spamudsendelse er pt. ganske enkelt et spørgsmål om at have et program til rådighed, der kan sende den ønskede meddelelse til de e-mail adresser, man har i en fil. Uden storhedsvanvid eller kendskab til et sådanne program, er det svært at tro, at det kan forvolde de store problemer. Hvis man som programmør har styr på de tekniske finesser på hvordan en e-mail bliver sendt gennem Internettet, og samtidigt ikke er i besiddelse af moralske skrupler, er der grundlag for at tjene nogle hurtige penge. Hvis et firma er villig til at trodse en given lovgivning, forsøge sig med spam og samtidig kan spore en fremgang i omsætningen af deres produkter, skulle man være en dum købmand, hvis man ikke kædede tingene sammen og gjorde et nyt forsøg. En produktudvikler og en forbruger af et produkt har brug for hinanden, forbrugeren vil gerne have det så billigt og effektivt som muligt. Udvikleren kan med et godt produkt, sagtens imødegå forbrugerens ønsker om dette. Et godt produkt giver en god omtale og dermed nye betalende kunder i butikken, hvorved udviklerens arbejdsindsats med at opretholde programmets effektivitet bliver betalt.

5.4 Intressenter

Intressenter må der givetvis være mange af eftersom spam er blevet et problem.

Personer, firmaer og organisationer, der kunne have et behov eller en interessei, næsten gratis, at kunne sende reklamer til millioner af mennesker kan karakteriseres efter følgende:

- Personer der vil have et bestemt budskab ud til et bredt repræsentativt udvalg af Jordens befolkning
- Personer der har en økonomisk interesse i at sende spam.
- Nyopstartede firmaer, der vil profiler sig selv.
- Firmaer der er økonomisk betrængte.
- Firmaer der har hård konkurrense.
- Firmaer der har et nyudviklet produkt.
- Organisationer der kan se fordele ved at slutte op om en sag.

5.5 Hvorfor er spam et problem?

Hvorfor kan folk ikke bare slette de få spam e-mail de modtager om dagen ?
Spam er netop et problem fordi at det er uønskede reklame e-mails man modtager. På Internetttet kan man ikke sætte et "ingen reklamer tak" skilt på sin internetforbindelse og ikke modtage reklamer. På internettet kan man kun prøve at filtrere ønskede e-mails fra, ved hjælp af programmer til selvsamme.
For langt de fleste modtager man kun et par spam mails om dagen hvis overhovedet nogen, og i disse tilfælde er det selvfølgelig ikke besværligt at trykke slet til dem fremfor at begynde at aktivt bekæmpe det.
Skulle man være så uheldig at ens e-mail adresse havner på en e-mail liste som spammer får fat i, begynder de få spam mails hurtigt at stige til mange pr. dag. Så begynder problemet at tage form. Hvis en modem-bruger, der betaler pr. minut online, skal hente 100 mails fra sin mailserver, og de første 99 er spam og den sidste er en vigtig mail som personen skal bruge. Det er nu brugeren selv der betaler for at modtage reklamer (spam) som han/hun ikke har ønsket at modtage. Udover at det koster brugere penge at skulle hente alle disse reklamer, er det både tidskrævende og unødig besvær.
Hvis man regner med at det tager en person ca. 2 sekunder at genkende en uønsket reklame mail, og derefter 2 seks til at slette den, kan man se tiden løber op. Ved 100 e-mails hvor 90 er spam bruger man så ca. 6 min på at slette ting man ikke har bedt om. Nu tænker folk jo "herregud 6 min er jo ingenting". Nej men får man så 100 mails om dagen hver dag hele tiden bliver man hurtigt irriteret og træt at at bruge disse 6 minutter på dette.

Wladyslaw [15] udtaler at han ikke anser problemet med en evt. overbelastning af mail serveren, som det største problem med spam, det kan løses med at købe noget bedre isenkram, han mener derimod det største problem er iritiationsmomentet med at åbne og kassere de spam mails der slipper igennem.
Nogle spammails indeholder et link, hvor det skulle være muligt at blive fjernet fra deres mailing list, men disse links virker sjældent, og det virker ikke fair at bede folk at at bruge tid på at afmelde noget de aldrig i sin tid har tilmeldt sig.

5.5.1 Er problemet stort?

Som en normal bruger der modtager et par spam mail om dagen ville det ikke være det store problem, men skulle man pludselig enden på en mailinglisten som spammere har adgang til, begynder det at tage fat. Generelt vil det udvikle sig til et problem efters om at folk i større og større grad vil modtage spam de ikke har bedt om. Det begynder også at være et problem og irritationsmoment for brugerne da de ved at bruge e-mail filtre og spam filtre udsætter sig for risikoen for at ikke modtage mail de vil have. hvis f.eks. en ønsket mail bliver sorteret fra af ukendte grunde.
Forbrugere der har oplevet at få spam, vil ofte føle sig nødsaget til at begrænse deres brug af e-mail addresser. Hvis man udleverer sin e-mail addresse et offentligt sted, f.eks en nyhedsgruppe, kan man føle angst for at den vil blive opsnappet og føjet til en spam mail liste. E-mailens funktion er hovedsageligt at fremme den elektroniske kommunikation, men spammere er gennem deres aktiviteter med til at begrænse den elektroniske kommunikation.
Ifølge Wladyslaw:

> ...er ca. 20 procent af Uiversitetes mailforsendelser, spam. Tallet skal tages med forbehold, for mange E-mails bliver ikke læst, men slettet via blacklister eller spamfiltre. Tallet varierer også meget, men det er mest op imod weekenden, at der indløber klassificerede spam mails

Grundet at e-mail er blevet en kommerciel handelsvare (lister med e-mails tjener folk penge på at sælge til spammere) eksploderer e-mail handel for tiden. Langt de fleste mennesker ender med at skulle tage forholdsregler for at undgå at deres mailbox bliver fyldt med reklamer de ikke vil have.
Da e-mail reklame er relativt gratis for spammere, er det langt den mest effektive måde at reklamere for noget til masserne. Man kan med en tilstrækkelig stor e-mail liste sende reklamer til f.eks. 2000 mennesker, hvor udgiften kun er det man bruger på at sende, altså typisk sin egen internetforbindelse. Hvorimod brugerne (modtagerne) selv skal "betale"for ressourcerne for at hente deres email. Derved koster det i realiteten modtageren af reklamen de penge som går i forsendelse.
F.eks. hvis en modembruger modtager 20 spam e-mails om dagen og betaler pr. minut han er på nettet, vil det koste ham penge at modtage disse e-mails, som

han i langt størstedelen af tilfældene ikke har bedt om at modtage. En sammenligning ville være at folk skulle betale for de reklamer de modtager gennem deres brevsprække, og IKKE har den mulighed at sætte “ingen reklamer tak” skilt på døren.
Dog er spamfiltre et skridt på vejen, til skiltet på døren, der afviser reklamer, men alle ved at en umoralsk person sagtens kan finde på at ignorere det skilt, og det samme gælder filtre. Spam kan sagtens slippe igennem, f.eks i forklædning.
Vi lavede i gruppen selv en test, for at se om vi ved at tilmelde en e-mail addresse til diverse mailinglister skulle “lykkedes” os at komme på en af de omtalte spam lister. Det gjorde det og vi modtog simpelthen e-mail reklamer fra afsendere som vi ikke havde bedt om at få noget fra.
Et fælles træk for disse e-mail var at de “retfærdiggjorde” deres egen udsendelse ved at skrive noget lignende *“du modtager denne e-mail fordi du har meldt dig til mailingliste hos vores partnerfirma”*. Vores in-box hos Hotmail [14]. blev uden oprydning og sletning af e-mails fyld til max antal på under 2 uger. Efter en oprydning, som skete efter 1 måneds forløb, havde vi ca. 413 e-mails liggende. Alle disse var uanmodede og ønskede. Nu lyder det ikke af så meget, men der skal tage i beragtning at vi ikke modtog e-mails i mere end 2 uger, pga vi efter den tid ikke havde plads til flere. Derfor er det rimeligt at gå udfra at vi efter al sansynlighed ville være offer for den dobbelte mængde hvis vores in-box ikke havde en maksimal størrelse.
Havde vi f.eks. skulle sidde med en modem opkobling til internettet, og kun tjekke vores e-mail (i dette tilfælde en hotmail konto) hver uge ville det tage utroligt lang tid at rydde op i ens in-box. Yderligere risikerer en person netop at in-box bliver fyldt, og e-mails sendt til adressen mens denne er fyldt kommer ikke igennem... altså kunne man direkte miste e-mail ved at modtage for meget uønsket spam.

Kapitel 6

Anti-spammers

6.1 Teknologibærere

Den teknologi der i dag er til rådighed til bekæmpelse af spam består hovedsageligt af anti-spam filtre, som brugerne selv installerer på deres computere. Mange af Disse fungerer ved at e-mails med bestemte karekteristika og opbygning bliver analyseret og klassificeret som spam eller ej. Det ville godt nok være mere effektivt at blokere spam på ISP siden, dog kan en ISP kun give en begrænset hjælp for at hjælpe brugerne med at undgå spam. Hvis ISP'ens filter er for strengt kunne de risikere at brugerne ikke modtager de e-mails som de beder om, og ligeledes hvis dette ikke er strengt nok, er det tvivlsomt hvor effektivt og nødvendigt dette er. Yderligere har de klient orienterede anti-spamfiltre en fordel, fremfor filtreringen hos ISP, nemlig black- og whitelister. En bruger kan nemt vælge at modtage spam og alt andet, fra nogle bestemte e-mail adresser, som er på hans whiteliste og intet fra andre (blacklisten). Alle de afsendere, der hverken er på white- eller blacklisten skal så igennem en analyse for at finde ud af om de er spam eller ej.

Nogle af de mest anvendte programmer til bekæmpelse af spam er f.eks. Spamassasin (se. 9.1) og Mailwasher (se. 9.2), som begge er beskrevet i flere detaljer senere i denne rapport.

En af forkæmperne for anti-spam, Paul Graham [8], arbejder pt. på sit egen filter, som endnu ikke er udgivet til offentligtheden. Ifølge Paul Graham vil dette filter kunne fange langt de fleste spam mails, og få 0 eller ekstremt få falske positive. Falske positive er netop hvad ethvert spamfilter bør undgå. Hvis en bruger "mister" en mail pga. filtrering, altså hvis en legitim e-mail bliver filtreret fra. Brugeren må hellere modtage en spam for meget, end miste en legitim e-mail. Derfor må målet for filtrene både være effektivt at sortere spam fra, og samtidig undgå at brugeren mister e-mails pga sorteringen.

6.2 Hvem bekæmper spam?

Bekæmpelse af spam sker af organisationer og personer. Paul Graham er en af disse personer som stiller spørgsmålet om hvad man kan gøre for at slippe for spam.Han vil gerne selv slippe for at modtage spam, og i processen udvikler han så et filter til at filtrere uønsket e-mail fra og kun modtage "rigtige"e-mails. Langt de fleste filtre er gratis, og i værste fald add-ware (programmet viser reklamer, når det kører). Dog kan man så betale skaberne af programmet en mindre sum penge, for at få en fuld ureklameret version af programmet. Mange filtre udvikles også under konceptet open-source, hvilket i princippet (afhænger af licensen) betyder at alle skal have adgang til kildekoden og frit må ændre i den selv. Yderligere betyder det også at folk ikke må tjene penge på programmer bygget på en sådan kildekode. Nogle vil måske bare gerne tjene lidt penge på at udbrede netop deres program, men langt de fleste går efter det lidt ædle formål at hjælpe andre. Udover at kunne anskaffe sig nogle programmer til at bekæmpe spam, findes nogle specielle websites der indeholder oplysninger som folk kan bruge i deres individuelle kamp imod spam. F.eks. kan de hjælpe brugeren med tips og tricks omkring at holde ens e-mail adresse væk fra såkaldte spam lister. En af disse er f.eks. den danske side Fabel [6]. Herunder siden kan man enddog benytte deres online spam filter, til at filtrere sine e-mails. De arbejder yderligere, frivilligt, for at hjælpe folk med at undgå spam, og bekæmpe udsendelse af spam. Tilsvarende findes i den internationale side spam.abuse.net [2] som også har samme formål, at stoppe spam.

6.2.1 CAUCE

CAUCE [4] [1] er en anden anti-spam organisation som nu, efter vores opfattelse, regnes som en af de højttalende stemmer i kampen mod spam. Den startede ligeledes som de fleste anti-spam sites, og organisationer, som en newsgroup der hovedsageligt følte at spam, man ikke har bedt om, er forkert. De sammenligner med at modtage reklamer på "modtageren betaler"princippet.

Det er 100 procent frivilligt arbejde, der hovedsageligt er drevet af folk der gerne vil slippe for spam, og have nogle værktøjer til at bekæmpe folk der udsender spam. De arbejder på at få lavet lovgivning der forbyder spam, og som holder i praksis. Dette er i sig selv en stor opgave. En af hovedaktørerne bag CAUCE er også en af de hovedansvarlige for spam.abuse.net. Som de selv beskriver sig selv:

> "We have no money, we have no office. CAUCE is a creation of the Internet – it exists on this web site, in newsgroups, discussion lists, and in the ideas and dedication of those who oppose the damaging, costly, and unfair practice of unsolicited commercial email."

[1] The Coalition Against Unsolicited Commercial Email

Det er en non-profit organisation der føler det er forkert at modtage spam e-mails. De skelner alle mellem to modeller for reklame e-mails.

- OPT-IN
- OPT-out

Opt-In er modellen som de foretrækker hvor brugeren selv aktivt skal tilmelde sig reklamer f.eks. på respektive firmaers hjemmesider eller lignende. Denne metode bruges f.eks. til mailing lister.
Opt-Out derimod er den uønskede model, som desværre er hovedsageligt aktiv i øjeblikket. Denne model specificerer at brugeren ikke selv beder om at modtage reklamerne, og derved retmæssigt modtager reklamer de ikke ønsker.

6.3 Politiske organisationer

Det er svært at undvære det elektroniske kommunikationsmiddel for brugerne af Internettet. De forskellige virksomheder udnytter det groft for at reklamere deres produkter selvom forbrugerne siger "reklame nej tak". DAF [2] af Internettet har klaget til Eu's parlament om at forbyde virksomheder som reklamerer for deres varer via elektronisk post. Så det er nu blevet forbudt for virksomheder at sende spam mails til de enkelte forbrugere, men problemet i det er at der ikke er kontrol over hvem der sender spam. De enkelte virksomheder opretter falske e-mail adresse, og sender stadig en del spam ud til forbrugere som ikke kan gør noget ved det.
Forbrugerombudsmanden Hagen Jørgensen sørger for at de forskellige virksomheder ikke bruger urimelige, vildledende, mangelfulde eller forbudte salgs reklamemetoder til at sælge varer. Han har sagsøgt mange flere virksomheder der overtræder markedsføringsloven. Han mener at det er groft at sende uopfordret reklamer via elektronisk post. Han har lavet en hjemmeside for forbruger der er utilfredse med spam. Hagen Jørgensen får ca. 100 klage mails om dagen af de utilfredse forbruger angående spam. De fleste klagede om udlandske virksomheder der sender spam mails men der blev også sendt mange fra danske virksomheder selvom DAF sørgede for at det blev forbudt. Forbrugerombudsmanden kører sag mod et dansk firma og der forventes at virksomheden for en bøde på over 100.000 kr. for at sende uopfordret spam mails til forbrugerne.

6.4 Spam communities

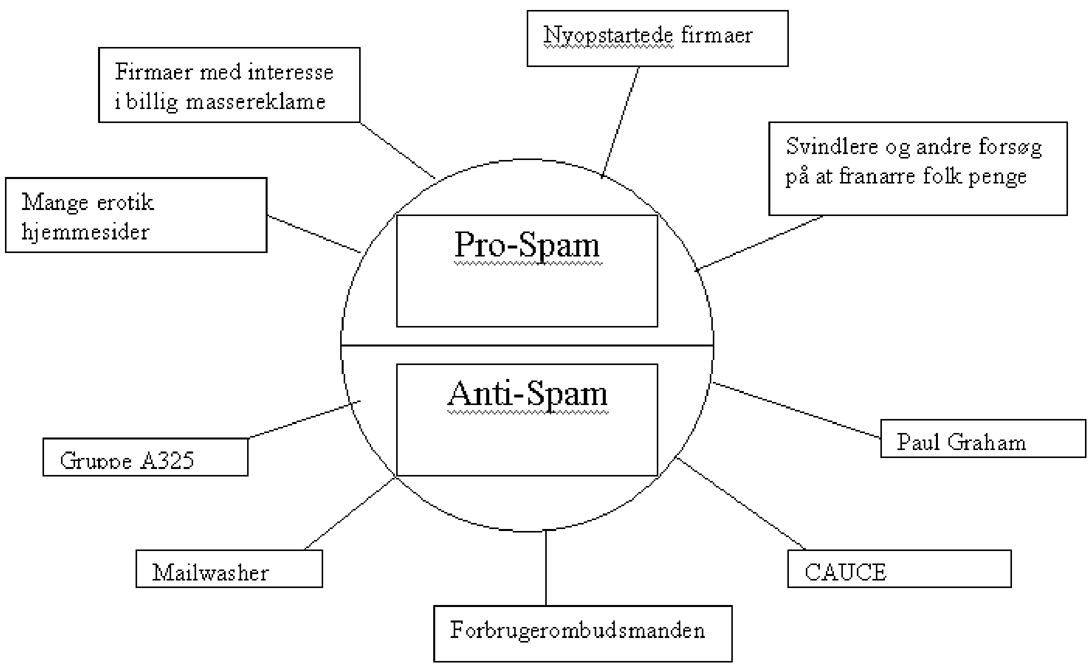
Der findes mange hjemmesider hvor der er hjælp at finde til spam ofre. De har lavet links, værktøjer osv. til at holde spam væk fra mailserveren. De fleste hjemmesider beskriver også hvordan spammeren får fat i ens e-mail adresse.

[2] Danmarks Aktive Forbruger

Fabel (for ansvarlige brug af e-mail) er blandt andet en af det mange der modarbejder spam. Deres formål er: (1)Udbrede kendskabet til ansvarlig brug af e-mail. (2)Påvirke lovansvarlige og ansvarlige i private eller offentlige virksomheder. (3)Støtte og om muligt deltage i undersøgelser omkring e-mail reklamer. (4)Hjælpe ofre af uønsket e-mail med råd og vejledning.
Fabel har lavet en test på danske mailservere, hvorpå af det 13.000 der er i Danmark er der 5500 der får spam. Det er ikke alle der er klar over det. Dem som er indstillet på at løse problemet, laver rettelser og vil gerne have fabel skal checke deres mailserver igen. Spam.abuse.net [2] og ZIP har hjælp til spam ofre, hvor der bliver beskrevet hvordan de skal filtrer og blokere for spam.

6.5 Spammers vs. Anti-Spammers

En model over de to sider der henholdsvis repræsenteres af spammers og antispammers findes i 6.1. Vi har folk såsom Paul Graham og udviklerne for anti-spam programmer og på den anden spammerne selv. Øverste halvdel af cirklen repræsenterer spammere, hvilket vi har omtalt til f.eks. at være nyopstartede firmaer, folk som prøver at narre penge fra folk, men alverdens planer. På den anden side har vi så bekæmperne af spam, såsom Paul Graham, udviklerne af spam filtre og lignende.



Figur 6.1: Oversigt og de 2 lejre. Spammers øverst vs. Anti-Spammers nederst

Kapitel 7

Lovgivning

Vi har objektivt belyst situationen omkring fænomenet spam. Men hvor står lovgivningen i dette. Kan man gøre noget aktivt ved spammerne, og fungerer det? I følgende afsnit ser vi på den nuværende lovgivning omkring spam. Det følgende er baseret på [24] og [13]. Der er en lovgivning om markedsføring i Danmark, den inkluderer også internettet. Den står under §6.a i markedsføringsloven og hører til under Forbrugerstyrelsen.
På nuværende tidspunkt findes der også en betænkning i justitsministeriet om It-kriminalitet, og det er en form for "noter" som der er brugt mange timer på at gennemtænke og som første bliver bruger ved de første retssager til at lave danne en straffelov mod om It-kriminalitet. Men første skal vi lige kigge på før omtalte §6.a som omhandler "Uanmodet reklame" det vil sige at man ikke har bedt om reklamer fra den der sender dem. Loven er fra den 1. juli 2000 hvor den trådte i kraft og bestemmer hvordan erhvervsdrivende må reklamer uanmodet på Internettet og hvad de må bruge til dette, og hvornår de må henvende sig til potentielle kunder.
Loven nr. 213 af 31 maj 2000 er trådt i kraft den 1. juli 2000. den omhandler ændringer af forbrugeraftaler og ændringer af markedsføringsloven § 6. a.

7.1 Markedsføringslov §6.a

Den gør sig gældende i alle uanmodet reklamer med direkte markedsføring til potentielle kunder, og det gælder alt lige fra fast ejendom til tjenesteydelser. Med undtagelser af aviser og blade medmindre at man reklamere kun med formålet at reklamer (at gør opmærksom på sig selv) Og det gælder også nyhedsbreve hvor man mest reklamer for egne produkter. Selv om man er fast kunde gælder §6.a stadigvæk, det er dog tilladt at sende nyt telefonnummer, hvornår varen ankommer til kunden eller noget af denne type breve, der skal siges her at det er ikke tilladt at prøve at omgås denne lov ved at tilbyde kunden en "bredere"vare. Henvendelser til en eller flere hvor der bliver brugt personlig oplysninger (også mail. adr.) er omfattet denne §6.a. og er ikke tilladt, uadresseret forsendelser er ikke omfattet denne §6.a. men mail adresser er netop ikke adresseløse.

7.2 Forbud mod anvendelse af elektronisk post

Elektronisk post er ikke kun e-mail, men også sådan noget som sms-beskeder og en hver form for videresending af disse eller opfordring til videresending er en overtrædelse af markedsføringslovens § 1. Hvis man sender et elektroniske postkort fra et firma skal man passe på hvis der er reklamer på det elektroniske postkort det kan være en overtrædelse af forbudet. En anmodning eller samtykke om modtagelse af reklame kan kun ske hvis personen klart og uden skyggen af tvivl ved hvad denne giver sit samtykke til, det betyder at alle standardvilkår eller forretningsbetingelser ikke lever op til dette medmindre at det er fremhævet på en særlig måde over for personen. Det kan virke lidt problemflydt, hvis f. x at forbruger lokkes via tilbud om at deltage i forskellige konkurrencer eller linende, fordi at reglerne om at give personlige oplysninger videre er lempet, så at kundeoplysninger må gives videre medmindre at personen gør indsigelser imod dette, men dette må ikke betragtes som en samtykke til at modtage elektroniske reklame, dette gælder derfor også spammails.

7.3 Andre direkte henvendelser

Man kan til en hver tid frabedt sig at modtage reklamer af en hver art og hvis man få dem alligevel fra den kilde/firma man sagt nej eller frabedt sig dette er det en overtrædelse af loven, for erhvervsdrivende må ikke rettet henvendelse af nogen art, hvis det er med henblik på at afsætte en vare eller ydelser når dette er frabedt. Men kan frabede sig disse direkte henvendelser ved at give den erhvervsdrivende besked herom, eller ved at give folkeregistret i den kommunen man bor i besked om at man ikke ønske henvendelser der er reklamer. Hvis man benyttet den sidste mulighed må den erhvervsdrivende ikke give oplysninger om personen videre, når det har markedsføringsformål. Det er ulovligt at opkræve betaling fra en person der frabeder sig henvendelser at man tilbagekalde sig tidligere samtykke Denne bestemmelse beskytter private forbrugere, offentlige myndigheder og da forbrugerombudsmanden ofte har modtaget klager fra mindre erhvervsdrivende er loven også kommet til at gælde for erhvervsvirksomheder som er drevet i personlig form.

7.4 Pligt til kontrol

Erhvervsdrivende har pligt til at kontroller om man har frabedt sig direkte henvendelser på en liste der bliver udarbejdet hvert kvartal af folkeregistret, med

mindre de på forhånd har modtaget en anmodning om dette, det gælder også for elektroniske post. Listen må kun videregives internt mellem erhvervsdrivende i erhvervsorganisationer.
Når en erhvervsdrivende henvender sig til en eller flere personer for første gang, skal han (hvis personen ikke står på før omtalte listet) gøre det helt klart og forståeligt for personen, at han har retten til at frabede sig alle henvendelse i fremtiden og den erhvervsdrivende skal på en nem måde gøre det muligt for personen at frabede sig dette. Disse rettigheder skal være iøjnefaldende og ikke gemt væk med små bogstaver bageste i henvendelsen.

7.5 Forbrugerombudsmandens beføjelser

Det er nu helt klart at fremsending af reklamer ikke må ske uanmodet heller ikke i form af elektroniske post (mails). Det har hele tiden være anset for dårlig markedsføringsskik at sende reklamer i form af elektroniske post fordi det påfører modtagerne en mere omkostninger og kan frarøve meget tid fra modtagerne. P.G.A. begrænsede ressourcer bliver forbrugerombudsmanden at prioritere så at der bliver taget hensyn til forbrugerne og derfor kan erhvervsdrivende ikke regne med hjælp fra ham, men selv anmelde overtrædelser af § 6.a til politiet Ved klare overtrædelser af §6.a vil forbrugerombudsmanden have mulighed for at gøre brug af den nye påbudsmulighed i markedsføringslovens § 19, stk. 2.som lyder

> "Forbrugerombudsmanden kan meddele påbud, hvis en handling klart er i strid med loven og ikke kan ændres ved forhandling."

[13] vil sagen blive videregivet til politiet.
Hvis erhvervsdrivende er i tvivl med hensyn på om et markedsføringstiltag evt. ville overtræde § 6.a. eller andre paragraffer kan de ifølge af § 18 bede forbrugerombudsmanden om en forhåndsbesked med hans syn på deres tiltages lovlighed.

7.6 Tiltag til at få kontrol over elektroniske markedsføring

Forbrugerombudsmanden har forsøgt i 1998 at forhandle et sæt retningsline med relevante branche og forbrugerorganisationer, men dette mislykkes blandt andet fordi man ikke kunne blive enige om brugen af elektroniske post til at reklamere med. Herefter udsendte de nordiske forbrugerombudsmænd en anbefaling som hed "fælles holdning til handel og markedsføring på Internettet og i tilsvarende kommunikationssystemer"som gik ud på at erhvervsdrivende kun skulle sende markedsførings- materiale via e-post og lignende, når de havde fåret lov til dette.

7.7 Første retssag

Informationerne kommer fra [3]
Udtalelser for firmaet kommer fra Fonns markedschef for Norden, Gjert Tolås. Forbrugerombudsmanden har i denne sag haft 150 klager over softwarefirmaet "Fonn Danmark" der har udsendt ulovlige reklame-e-mail (spam) og derfor meldt firmaet til politiet.
Det er første gang i Danmark at en domstol har en sag om spam-mails og der p.t. kun en betænkning i justitsministeriet som vil bliv taget i brug ved de første retssager om It-kriminalitet også på den måde danne grundlag for en straffelov. Sagen skal for domstolen (som er Sø - og handelsretten) i december/ januar og har taget et år at forberede. Fonn Danmark er anklaget for at have udsendt telefax og e-mails til folk der ikke kendte til firmaet. Hvis de bliver dømt kan de blive idømt en bøde, hvor størrelsen på bøden vil danne præcedens for fremtidige sager. Firmaet mener at de er blevet uretfærdig behandlet i Danmark, fordi de ikke mener at de bevist har henvendt sig til private forbruger og kun henvender sig til virksomheder, (men i følger §6. a. gælder denne også for mindre virksomheder), Firmaet er blevet kaldt Danmarks største spammer og de siger selv at de kun har udsendt til ca. 10.000 - 15.000 adresser og de forstår ikke logikken i at før sag imod dem når man samlinger med de mange spam- mails fra udlandet. Markedschefen erkender at have udsendt de mange mails selv om han kendte til den danske lovgivning, på trods af de mange klager der siger noget andet siger markedschef at han ikke har udsendt mails siden april 2002, hvor han havde en dialog med forbrugerombudsmanden, og at de seneste klager må have været gamle mails.
markedschef finder det urimeligt at man efter 14 års med nyhedsbreve har holdt kunderne opdateret og nu skal man til at spørge om de vil have reklamerne, han siger ydre mere at det vil irritere kunderne mere at blive ringet op, for at høre om det vil have deres reklamer og det bliver alt for dyrt for firmaet. Alt dette har ført til at der er skrevet mange sider på nettet hvor de hakker ned på firmaet og advarer imod at de sender spam det sammen sker hvis man laver en søgning på markedschefen Gjert Tolås, som tager alt reklame som gode reklame, for han forstår ikke at kunderne ikke reagerer på samme måde. Firmaet havde overvejet at udvide med en telemarketingsafdeling i Danmark men er nu blevet nervøs for retssystemet i Danmark. Der opfordres stadigvæk til at træde frem hvis de har modtaget "spam" fra "Fonn Danmark" siden i sommers.

7.8 Regulering af spam via lovgivningen

Vores lovgivning i Danmark bliver ikke på nu værende tidspunkt overholdt, det fremgår tydeligt af artiklen ovenfor, hvor Gjert Tolås siger at han var bekendt med den danske lovgivning § 6.a. i markedsføringsloven, men alligevel sendt mails ud, vi kan også se at den danske lovgivning ikke bliver overholdt på de mange

spammails som kommer fra udlandet. Vi ikke kan gå ud fra at de har samme lovgivning eller, at de evt. helt mangler en lovgivning på området med markedsføring og derfor er ligeglade med vores lovgivning.
Vi kan derfor ikke håndhæve vores lovgivning i alle lande, medmindre at vi får dem til at vedtag den danske lov § 6.a. i markedsføringsloven i disse lande og at de efterlevet denne. Vi kan se, om vi får et godt grundlag til vores kamp med at bekæmpe spam, når der er faldt dom i retssagen imod Fonn Danmark som starter i dec.2002/jan.2003.det kan blive alt betydende for fremtidens bekæmpelse mod spam om politiet får nogle gode redskaber (i form af bøde og andre form for straffe) mod lovovertræderne.

7.8.1 Wladyslaws mening om lovgivningen

Wladyslaw [15] har udtalt følgende omkring lovgivningen:

Jeg har den opfattelse af det danske retsystem, at det er relativt langsommeligt. Vores lov på spam området er ikke optimalt nok til at klare situationen. Situationen er som føleger: man kan klage til den rette instans evt. ombudsmanden, over man modtager meget spam fra en bestemt ISP, man skal så opgøre problemets omfang enten i tid eller penge, hvilken værdighed har fx 20 min?, sagen er død før den starter. I USA er det lettere, spammerne får en bøde pr. sendt e-mail.

Kapitel 8

Løsnings forslag

Vi vil dele vores løsningsforslag op i følgende kategorier da, disse er væsentlig forskellige fra hinanden. Det skal også pointeres at her kun er repræsenteret nogle teknikker, og der findes rigtig mange løsningsforslag.

8.1 Løsning med en fremtidig lov

Det er efter vores opfattelse at hvis man kan efterleve den lovgivning der er p.t. § 6.a. i markedsføringsloven, og når der så er vedtaget en straffelov (udfra den betænkning der er p.t. og de første par sager som der skal til dette), at så er vi rimeligt rustet mod spam rent lovgivningsmæssigt. Men der skal mere oplysning til forbrugerne så de er helt klare over deres "nej tak"mulighed, også klager forbrugerombudsmanden når de får en spam så det bliver så dyrt som muligt at udsende spam. Hvis den markedsføringslov ikke kommer til at fungere har vi svært ved at se andre lovløsninger.

8.2 Tekniske

Tekniske løsningsforslag er der mange af. Man kan bygge eller installere et af de mange eksisterende filtre. Forskellige teknikker til frasortering af spam e-mails er bl.a. blocking lister, filtre bygget på bayesiansk sandsynligheds regning med tekstgenkendelse. De førende filtre på markedet anvender faktisk begge disse teknikker. Paul Graham [8] har et meget vigtig argument, nemlig at spam mailens achilles hæl er dens indhold. Så længe spammeren har det formål at sælge et produkt, vil indholdet af spam mailen automatisk være genkendelig for et filter bygget på tekstgenkendelse. Spammeren vil virkelig blive nødt til at ændre sin besked væsentlig meget for at slippe gennem et af de nuværende filtre. Løsningen vil måske være for spammeren at kamuflere sin besked som et “uskyldigt” brev næsten kun indeholdende et link til hans firmas hjemmeside. Denne spam vil således være svær at komme af med..

En anden teknisk løsning der ikke handler om at komme spam til livs er brugen af midlertidige eller tidsbestemte e-mail adresser. Der er generelt kun 2 måder ens e-mail adresser kommer på en såkaldt spam liste. Enten har du selv delt den ud eller også har der været en såkaldt "mail address harvester" der har opsnappet din e-mail adresse i forbindelse med en publikation i en nyhedsgruppe el. lignende. I forbindelse med nyhedsgrupper ønsker man ofte at en publikation bliver besvaret i en vis periode, dvs. at svar der kommer et år senere måske ikke har nogen interesse. I denne forbindelse kan man med succes anvende midlertidige e-mail adresser. Der findes firmaer, f.eks Mailtraq [1], der tilbyder tidsbestemte e-mail adresser. Disse kan så bruges til at åbne en diskussion i f.eks. 1 måned. Derefter er diskussionen ikke relevant. Skulle det så ske at adressen ender på en spam liste er skaden ikke så stor.
En måde at bekæmpe "mail address harvester" bots på er en såkaldt giftig pille. Man opretter en tilsyneladende normal del af sin hjemmeside. Den skal indholde en del til meneskelig besøgende om at de blot skal ignorere denne del af hjemmesiden. Ligeledes skal der laves tags der informerer portal botter som google-bot om denne side ikke er reel. Siden skal laves i et script sprog som kan generere lignende sider med e-mail adresser, som tilsyneladende er reelle nok, men ikke har nogen faktisk modtager, på siden skal man også sørge for at lave et link til endnu en falsk e-mail side. Vi vil ikke i detaljer komme ind på hvordan dette gøres, mere information kan findes på [26]. På denne måde kan man altså fange en harvester bot, således at den spilder sin tid og ikke får rigtige e-mail adresser. Dette hjælper også til at bekæmpe spammmeren der har botten kørende da han kommer til at sende en hel masse spam mails forgæves.

8.3 Organisatoriske

Vil man virkelig spam til livs, må man granske lidt i grunden til at spam i det hele taget får lov til at eksistere. Spam er en billig måde at reklamere på, så billig at de firmaer der bruger spam reklame, ofte kalkulerer med at en lille brøkdel af modtagerne læser og rent faktisk svarer på spam mailen. Man kan så spørge sig selv om hvor lidt svar skal spammeren have for at det kan betale sig at udsende spam? En organisatorisk løsning vil ganske givet være at organisere kampagner der opfordrer folk til at boykotte spams. En passiv løsning er langt bedre end en given aggressiv løsning. Man kunne forestille sig at en person bliver så træt af at modtage spam mails, at han aktivt begynder at sende klager tilbage til den der udsender spam e-mails. Dette kan kun frarådes, da spammeren ved at vedkommende modtager og læser disse spam mails. Spammeren der kun kan eksistere så lang tid han modtager bekræftelser på hans spam mail, vi kun intensivere fremsending af spam mails til denne person, da han mærker en potentiel kunde. En del af løsningen vil klart ligge i at opfordre folk til at filtrere sin e-mail, således at de ikke modtager spam mails og derfor ikke kan bekræfte spammeren i at hans mails kommer frem og bliver læst og taget til efterretning. Dog kan spammere sansynligt ofte bare udsnede deres spam, og derefter glemme den konto ham brugte til udsendelsen. Dette kan jo også være for at undgå at folk f.eks bekæmper ild med ild, og sender hans spam mails tilbage, eller lignende angreb.

Kapitel 9

Spamfiltre

Vi har efter en gennemgang af forskellige løsningsmuligheder, valgt at lave en research på nogle eksisterende spamfiltre.
Vi har valgt at beskrive 3 forskellige spamfiltre. Vi har som det første valgt spamassasin 9.1, da det er Aalborg Universitets udvalgte spamfilter. Mailwasher 9.2 har vi valgt som nummer 2, da det er et filter der har været på banen i relativt lang tid, og burde være godt gennemprøvet, Bogofilter er vores valg nr. 3. det er et forholdsvist nyt spamfilter, men baseret på nogle teknikker, fremlagt af Paul Graham, som vi mener er en spændende nytænkning inden for spamfiltre.

9.1 Spamassasin

Spamassassin er navnet på et mailfilter som er lavet til at identificere spam. Dette sker ved brug af et mønster-baseret regelsystem som programmørerne har lavet meget detaljeret og det er derfor nødvendigt at det bliver opdateret regelmæssigt for at bevare effektivitet af filteret. Filteret er blevet testet meget bredt på mailens header og body.
Spamassassin er gratis i Unix-udgaven og kan hentes som open-source, mere information om open source kan findes på www.opensource.org. Der findes en kommerciel udgave til integration i MS Outlook, som udbydes af Deersoft.
Der er 4 punkter som Spamassassin bruger til at identificere spam på:

"Header analyses": Spammer prøver at skjule sig bag en masse tricks, og disse tricks prøve spamassassin at gennemskue ved at se i mail-hovedet f.eks. at IP-adressen er fra 0-255 for at være rigtig og evt. stavefejl [30] nederst side 116. f.eks. at IP-adressen er 306.205.121.012 så er det en falsk mailheader fordi at IP-adressen 306 er større end 255, det kan også være at et ord er stavet med et stort bogstave i stedet for et lille bogstave. “ Form” er falske fordi det staves “form”, på denne måde kan man sige med rimlig sandsynlighed at det er Spam, hvorfor ellers prøve at ændre i headeren.

"Tekst analyse": De fleste spammails har en speciel karakteristisk opbygning. Mails der mangler karakteristisk opbygning kan være præget af at skille sig ud fra (f.eks.hvordan et alm. brev er opbygget). En anden måde mails kan skille sig ud kan være på enkelte ord eller på ordsammensætninger (f.eks "Free" er et ord der er meget brugt i spammail´s eller sammensætning "Free money") ,og disse kan SpamAssassin genkende.

"Blacklists": Spam Assassin gør brug af flere store kendte blacklister f.eks Mail-abuse.org[2] og ordb.org[20], men den egentlige dominerende liste er den der bliver lavet i samarbejde med Razor.

"Razor": Razor også kendt som "Vipul´s Razor"er et netværk der aktivt bekæmper spam, og dette gør de ved at opdaterer servers databaser med nye bekræfte Spammail´s og derved blokere for "mail.en", og skære dem væk, og her igennem at opdaterer nysammensætninger og opbyggeningen af mail´s. Denne mail bruges til at vedligeholde den orddatabase som Spamassassin er bygget på. Der udover prøver de i sammenarbejde med andre at spore spammerne på deres signaturer i mail´s fordi de sender mange ens mails af sted på samme tid. De lader en mail ryge igennem og blokere alle mails der er lig med denne.

SpamAssassin har en tests side hvor man kan se i hvilket område af mailen der er testet og hvor i verden mailen er kommet fra, kort beskrivelse af testen, testens navn og hvor meget den har scoret i SpamAssassin efter standard indstillingerne. Testen er lavet ud fra deres koncept hvor man selv kan tilfører sine egne indstillinger og der med at gøre SpamAssassin mere personlig. For mere information kan der henvises til SpamAssassins hjemmeside [27]

9.1.1 Aalborg Universitet og Spamassassin.

[15]
Aalborg Universitet bruger spamfiltret "Spamassasin", på det tidspunkt de anskaffede sig deres spamfilter, var Spamassasin, det eneste der virkede godt. Wladyslaw udtaler at det er forbundet med besværligheder, at bestemme hvad der er spam og ikke spam. Universitetet har valgt at bruge spamfiltret Spamasassin. Spamfilteret fjerner alt hvad Spamassasin betegner som spam. Det er meget resourse krævende, at checke mails igennem spamfiltret, det er derfor vigtigt at have en god og opdateret blackliste. Universitetes mailserver er en dual processing [1] maskine, den laver ikke andet end at ekspederer mailes videre, det er ikke forbundet med problemer, men bliver serveren bombarderet med spam, kommer de 2 processore på arbejde, for alle mails skal igennem spamfilteret. Mange ISPer får betaling pr. gennemsendt bytes og det er derfor problematisk at få dem aktivt

[1] Computer med 2 CPU

motiveret, til at stoppe spam, derfor er det nødvendigt med et spamfilter som spamassasin. Universitetet har været nødt til at droppe Spamassasins Whitelist funktion, for at forebygge retsager mod udenlandske ISPer. En effektiv måde at stoppe spam fra udlandet på, er med Spamassasins sprogfunktion, hvor man kan installere div sprog, hvorefter programmet, går ind og vurderer om mailen passer til den omgivelse den er i, dvs. om der findes specielle fremmede ord eller tegn, der mistænkeliggør mailen for at være spam, ex. en mail på kinesisk, afsendt til Danmark hvor afsenderen er en virksomhed.

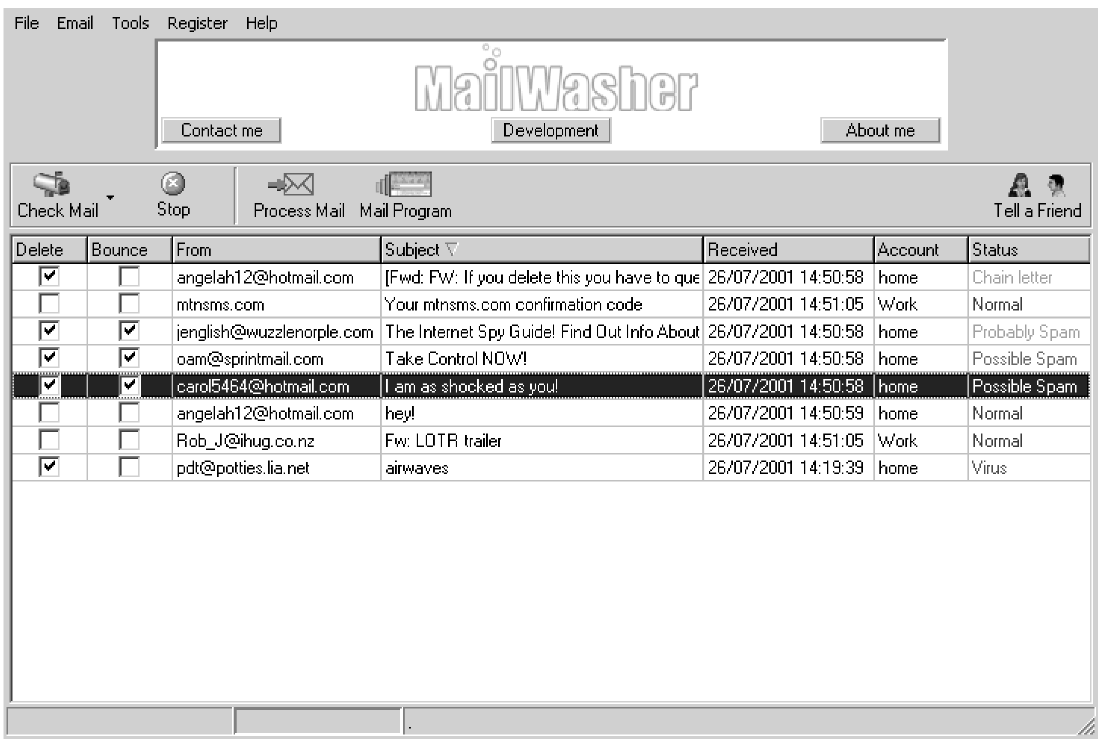
9.2 Mailwasher

Mailwasher [28] er et spamfilter, der er blevet rost i de medier for sin effektivitet. Filteret er virksomt på ens mail server. Indkommende mail til mail serveren bliver gennemgået af filteret,der efter bestemte kriterier, som suverænt er bestemt af klienten, kategorisere den indkommende mail efter de 3 følgende afsender kategorier:

- Whitelist, dvs. en venneliste. Disse folks e-mail adresser skal indtastes enten som et enkelt navn fra et domæne, eller med et wildcard hvis det fx. er alle folk fra et domæne, som man anser som sine venner. Et eksempel på brug af et wildcard kunne være *@but.auc.dk. Hvis afsenderens e-mailadresse står på whitelisten, passerer mailen Ukritisk igennem spamfilteret og står klar til at nedtage.
- Blacklist, dvs, en uvenneliste. Eftersom spam har været et voksende problem igennem længere tid. er det klart at der eksisterer optegnelser over e-mail adresser, der tidligere har sendt spam. eller over åbne servere, der ukritisk sender e-mails videre. Denne liste kaldes en blackliste. Mailwasher har som et tilbud, tilkoblet, ORDB, VISI og SpamCop Hvis afsenderens e-mailadresse står på blacklisten, sendes mailen enten tilbage til afsenderen, eller slettes. Hvis e-mailen har benyttet en server, der er registreret som åben, vil den ikke nå frem til klientens mail server, da mail serveren ikke vil modtage den.
- Ubekendt afsender, dvs. afsenderens e-mailadresse er ikke i blacklist eller whitelist kartotekerne. Spamfilteret gennemgår nu selve teksten. En optegnelse over elementer indeholdt i en e-mail, der indikerer at der er tale om spam er indlagt i mailwasher, men er dog utilgængeligt til gennemsyn. Hvis denne filtrering ikke sletter eller sender e-mailen tilbage, vil den blive vist på mail serveren.

Blacklister er et godt supplement til et spamfilter, men er ikke en nødvendighed, da de e-mails der kommer fra en Blacklistet adresse, burde blive klassificeret som spam af filteret og behandlet derefter. Blacklister aflaster spamfilteret for at filtrere de e-mails som Blacklisterne har sorteret fra. Blacklister bliver jævnligt

opdateret med alle nyopdagede e-mail adresser, der har sendt spam og registrerede åbne servere. Der eksisterer mange web sider der evt. i forbindelse med deres eget udviklede spam filter, er meget interesseret i at man sender en e-mail til dem, hvor man angiver hvad man har modtaget af spam og hvilken e-mail adresse, der har sendt den, hvorefter denne webside føjer e-mailen adressen til deres blackliste, dette er også en facilitet i forbindelse med mailwashers spamfilter.
Det deciderede spam filters fornemmeste opgave er at behandle den indkommene mail som klienten måtte ønske, dvs. at når filteret er opsat til at filtrere ens e-mail, efter indbyggede kriterier, skal der ikke indløbe e-mails, klar til at nedtage, på ens mail server, som man på forhånd har sorteret fra via div.. kriterier, filteret skal "luge alt ukrudtet væk". Filteret koncentrerer sig om hvilke ord e-mailen indeholder.Ord som free, offer, money osv. er jo klart ord, der er omspændt med faresignaler, hvis de indgår i en e-mail , det er jo sådanne ord en forretningsmand slynger om sig med, når der skal sælges eller vækkes interesse for et produkt. Sandsynligheden for om det er spam, ens indkommende mail server har modtaget er stor, hvis disse ord eller tilsvarende ord, der passer til denne karakteristik, er til stede.



Figur 9.1: Mailwashers interface

Brugen af Mailwasher er relativt nemt, man opstarter programmet ved at dobbeltklikke på programmets ikon, hvorefter optegnelse over ens e-mails på ens

e-mail server vil blive vist. Man kan så gennemse denne optegnelse og bestemme sig for, hvilke e-mails man vil have downloaded ned på ens computer, de frasorterede kan man så enten sende tilbage til afsenderen eller slette. Hvis man vælger at sende den tilbage til afsenderen, vil Mailwasher bruge en uoplyst algoritme til at bestemme den hurtigste rute, via ens egen Mail server, tilbage til afsenderen.
Hver gang man egenhændigt afviser en e-mail, altså enten sletter den eller sender den tilbage til afsenderen, bliver e-mail adressen automatisk blacklistet og hvis denne adresse dukker op igen en anden gang, bliver den automatisk slettet eller tilbagesendt. Hvis man gentagende gange får spam afsendt fra samme domæne, men med forskellige navne, er der en facilitet i Mailwasher, der gør det muligt, at indsætte et "wildcard"et eksempel kunne være, at man har modtaget spam fra fx. hugo@mailsp.com og peter@mailsp.com. Man kan så blackliste alt fra dette domæne ved at skrive *@mailsp.com.
Omfanget af de blacklister man ønsker installeret på Mailwasher, er helt op til en selv, man kan købe sig til store blacklister, eller finde nogle gratis. Det er kun ens pengepung og operativsystem der sætter grænsen, Windows 95, 98 og Me, har en fastsat grænse på 32,000 tegn indeholdt i en liste.

9.3 Bogofilter

Bogofilter er forholdsvist nyt spamfilter som er lavet a Eric S. Raymond og er udviklet udfra ideerne fremsat af Paul Graham [8] i *a plan for spam*. Filteret er opensource og ligger direkte på nettet tilgængeligt for enhver. Bogofilter er designet og optimeret til at være et hurtigt baysiansk spamfilter. Det er opbygget af hurtige algoritmer skrevet direkte i programmerings sproget C og benytter Berkeley DB til hurtigt at kunne processerer e-mails. Filterets hurtighed er en væsenlig faktor for steder hvor der er store mængder af mail, og dermed er filteret måske knapt så meget henvendt til den normale bruger som de system administratorer som behandler store mængder af spam/mail.
Bogofilter er opbygget på den måde at det tager en standard email/tekst som input og checker denne tekst opmod en database af *ham* og *spam* ord og udregner en statistisk sandsynlighed for om den pågældende mail er en spam mail. Denne metode er forholdsvis simpel men viser sig at være meget effektiv, men for at forbedre på Paul Grahams fremgangs måde er Bogofilter forsynet med en smarter måde opdele teksten på, så bl.a. IP-adresser. og hostnames forbliver intakte hvilket forbedre filtreingen væsentligt.
Første skridt er at opdele input teksten i *tokens*, disse tokens repræsentere sammensætning af karakter fx. er xxx ikke et ord men en sammensætning af bogstaver som sagtens kunne forkomme i en spam mail. Denne mængde a tokens bliver alle checket i *ham* og *spam* ordlister som holder styr på hvormange gange de pågældende tokens har optrådt i forgående spam/non-spam mails. Efter at have udregnet hver enkelt tokens sandsynligheds værdi, udvælges et bestemt antal af

disse til at udregne den samlede sandsynlighed for at mailen er en spam-mail. Den samlede sandsynlighed er udregnet ved hjælp af “Bayes´s theorem on conditional probabilities”, hvis denne sandsynligheds værdi overskrider grænsen på 0.9, klassificeres mailen af Bogofilter som spam ellers bliver den klassificeret som non-spam.
Et par af de features i Bogofilter:
Når du svare på en mail bliver denne automatisk opdateret i *ham*-ordlisten, udfra filosofien at man ikke ville svare på en spam-mail, hvilket bare ville forøge spam mængden.
En feature er også at kunne klassificere spam mails der er kommet igennem filteret som spam, ved at slette mailen med den option at den bliver adderet til spam listen og dermed formindske sandsynligheden for at der kommer en ligende spam mail igennem filteret.
Ovenstaaende text er baseret udfra Eric Raymond’s site [16].

9.4 Blockinglister

Spamfiltre kan gøres mere effektive ved brugen af offentlige blacklister, som alle har adgang til, og alle kan bruge. På disse lister kommer f.eks. kendte spammere, som folk anmelder, så, og så kan andre folk undgå at få mail fra disse. Der findes nogle stykker af sådanne lister, som mange programmer inkl. Mailwasher 9.2 , og SpamAssasin 9.1 gør nytte af... Princippet er enkelt. De tjekker simpelthen afsender mod en sådan liste, og hvis den matcher blokeres afsenderen. Ulemperne ved sådanne lister er selvfølgelig hvor meget man lige kan stole på sådanne lister. Heldigvis er det sådan er skulle man uretfærdigt være kommer på en sådan liste kan man skrive til ophavsmanden af listen og rede trådene ud. En anden ulempe er naturligvis at hvis man i et tilfælde faktisk ønsker at modtage noget fra en person som er på en sådan liste. Man ville i sådan tilfælde aldrig modtage de e-mails og det behøver afsendereden ikke nødvendigvis at være klar over. Sådanne blacklister som alle kan bruge er meget gode værktøjer til hurtigt og effektive gøre en samlet indsats mod spam. Alle kan tilføje folk til listen, og derved hjælpe hinanden med at undgå spam fra selvsamme personer, og dette bruges meget ofte i spam filtre.
Et af problemerne med en sådan liste er naturligvis at folk ikke har den store chance for at opdage hvis de, uretfærdigt, står på en sådan liste. Yderligere kan dette skabe store problemer for f.eks. et firma hvis de skulle være kommer på listen ved uheld. Man kan sige at sikkerheden for at folk ikke bare tilføjer hinandens adresser, ikke er så stor, da denne form for blackliste er baseret på at folk henvender sig med spammernes e-mail adresse. GAZnet [7] er en sådan liste som Mailwasher 9.2 kan benytter sig af, hvor folk kan sende forfatteren af listen en e-mail med adresser over spammers, som han så tilføjer i filtret. Dog kan det ikke siges hvilke foreholdsregler forfatteren tager for at undgå at “uskyldige” folk

kommer på listen.
Ifølge promonente anti-spam folk, såsom Paul Graham [8] fungerer blockinglister som disse dog desværre ikke særligt effektivt. De fanger slet ikke så mange spammers som de burde, delvist pga at spammere sjældent bruger samme e-mail addresse. Yderligere er de besværlige hvis folk som ikke spammer, ved uheld ryger på en sådan liste, hvilket skaber problemer for afsenderen.

Kapitel 10

Bayesiansk filosofi

Vi vil her gennemgå filosofien bag opbygningen af et bayesiansk netværk, der kan udbygges til at udregne en sandsynlighed for hvorvidt en given e-mail er spam. udfra resultatet kan man rådgive modtageren af den pågældende e-mail, til om han skal smide e-mailen væk eller læse den. Hvis modtageren foretrækker det, kan denne proces automatiseres sådan at e-mails med en bestemt sandsynlighed bliver kasseret uden at han ser det.

Denne forklaring af bayesianske netværk er hovedsageligt bygget på [10]

For at forstå et bayesiansk netværk må vi tage udgangspunkt i sandsynligheds-regning. Hovedelementet i sandsynlighedsregning er en hændelse i vores verden, enten i fortid, nutid eller i fremtiden. Et klassisk eksempel derpå er et kast med en mønt, der lander med en side opad. Givet en hændelse h, er den mest gængse opfattelse at dens sandsynlighed er frekvensen af hvor tit den hænder, når man laver et eksperiment, hvor man kaster en mønt 2 gange, med mulige udfald h og dens modsatte $\bar{h}$. Man kan antage at sandsynligheden af h repræsenterer graden af hvorvidt en person tror på et bestemt udfald i *et* eksperiment. Han tilskriver så en værdi i intervallet [0;1] hvor 1 betyder at han er sikker på at h hænder. 0 betyder at han er sikker på at h ikke hænder, alt der imellem betyder, at han til en vis grad tror på at h vil hænde.

Opfattelsen af sandsynlighed kaldes også ofte objektiv eller frekventiel fortolkning. Hvorimod opfattelsen af graden af hvor vidt en person tror på h kaldes også subjektiv eller bayesiansk fortolkning, opkaldt efter Thomas Bayes. I bayesiansk fortolkning vil sandsynligheden altid være afhængig af personenens kendskabs niveau. F.eks. hvis man giver en person en mønt vil han nok sætte sandsynligheden for at et kast med mønten giver plat, til at være $1/2$. Hvorimod hvis vi nævner muligheden for ham, at mønten er så deform at den vil give krone, vil han nok tillægge den en større værdi. Dette kan illustreres således:

	mønt	deform	normal
side	-	-	-
plat	-	0.25	0.50
krone	-	0.75	0.50

$P(h|\xi)$ læses sandsynligheden for h indtræffer givet ξ er indtruffet. F.eks. $P(side = krone|mnt = deform)$ vist ovenfor. Denne model giver også mulighed for, at en person antager, at noget er rigtigt. F.eks. kan vores møntkaster antage at mønten giver plat den 11. gang, hvis den har givet plat de første 10 gange. Vi skriver $P(h_2|h_1, \xi)$ for sandsynligheden for h_2 hvor h_1 er sand og givet baggrund ξ.

10.1 Regneregler

Som eksempler på regneregler vil vi vise sum- og produktreglen. Fra disse kan mange andre regneregler også fremsættes.
For enhver hændelse h og dens modsatte $\bar{h}$ gælder:

$$P(h) + P(\bar{h}) = 1 \tag{10.1}$$

Den fundamentale sætning siger at, for hændelser h_1 og h_2:

$$P(h_1, h_2) = P(h_2|h_1) * P(h_1) \tag{10.2}$$

hvor $P(h_1, h2)$ er den simultane eller fælles sandsynlighed for h_1 og h_2. Denne gælder også for sæt af variable.
Fra nu af begynder vi at bruge variable. Variable skrevet som x betegner en enkelt variabel, mens X betegner et sæt af variable. Vi skriver $x = k$ for at tilskrive at x er i tilstand k. Når vi ser tilstanden af X kalder vi dette sæt af observationer, tilstand k af X og skriver $X = k$. Sandsynligheden for et sæt af variable X, skrevet $P(X)$, er sættet af sandsynlighed $P(X = k)$ for alle tilstande af X.
Før vi fortsætter må vi hellere få Bayes Theorem på plads:

$$P(X|Y) = \frac{P(Y|X)}{P(Y)} P(X) \tag{10.3}$$

hvor $P(Y) > 0$ Her er $P(X)$ den tilskrevne sandsynlighed for X, som også kaldes baggrundssandsynligheden for X, før vi kender Y, og $P(X|Y)$ er den tilskrevne sandsynlighed for X når vi kender Y.

Eksempel

Givet det generelle eksempel:

$$\begin{aligned} P(X) &= P(x_1), P(x_2) \\ P(Y) &= P(y_1), P(y_2) \end{aligned} \tag{10.4}$$

så er:

$$\begin{aligned} P(Y|X) &= \{P(Y=y_1|X=x_1), P(Y=y_1|X=x_2), \\ &\quad P(Y=y_2|X=x_1), P(Y=y_2|X=x_2)\} \end{aligned}$$

med 10.3 giver dette:

$$\begin{aligned} P(X|Y) &= \frac{P(Y|X) * P(X)}{P(Y)} \\ &= \begin{cases} P(X=x_1|Y=y1) \\ P(X=x_2|Y=y1) \\ P(X=x_1|Y=y2) \\ P(X=x_2|Y=y2) \end{cases} \\ &= \begin{cases} \frac{P(Y=y_1|X=x_1)*P(X=x_1)}{P(Y=y_1)}, \\ \frac{P(Y=y_1|X=x_2)*P(X=x_2)}{P(Y=y_1)}, \\ \frac{P(Y=y_2|X=x_1)*P(X=x_1)}{P(Y=y_2)}, \\ \frac{P(Y=y_2|X=x_2)*P(X=x_2)}{P(Y=y_2)} \end{cases} \end{aligned}$$

Vi har den generaliserede sum regel

$$\sum_Y P(X,Y) = P(X) \tag{10.5}$$

hvor $\sum_Y$ er en generaliseret sum. Dette kalde også for at marginalisere Y ud.

Eksempel

Hvis $P(X,Y)$ er:

	Y	y_1	y_2
X	-	-	-
x_1	-	$1/8$	$3/8$
x_2	-	$2/8$	$2/8$

så er $\sum_Y P(X,Y)$:

$$\begin{aligned} \sum_Y P(X,Y) &= \\ &= P(X, Y=y_1) + P(X, Y=y_2) = P(X) \\ &= \begin{cases} P(X=x_1, Y=y_1) + P(X=x_1, Y=y_2) \\ P(X=x_2, Y=y_1) + P(X=x_2, Y=y_2) \end{cases} \\ &= \left. \begin{cases} 1/8 + 3/8 = 1/2 = P(X=x_1) \\ 2/8 + 2/8 = 1/2 = P(X=x_2) \end{cases} \right\} = P(X) \end{aligned}$$

Endeligt er der ekspansions reglen:

$$P(X) = \sum_{Y} P(X|Y)P(Y) \tag{10.6}$$

Som bliver udledt ved hjælp af *den fundamentale sætning*10.2:

$$\sum_{Y} P(X,Y) = \sum_{Y} P(X|Y) * P(Y) = P(X) \tag{10.7}$$

10.2 Kausale netværk

Med udgangspunkt i [29] vil vi nu definere Kausale netværk vha. grafiske modeller. Gode eksempler kan findes i [29] i kap. 2.1 til og 2.1.5. Disse bliver uddybet i kap. 2.4. [29]

Et kausalt netværk består af et sæt variable og et sæt direkte sammenkædninger mellem variable. Når man taler om orienterede grafer bruger man familie relationer: hvis der er en pil fra A til B siger man at B er et "child" af A og at A er "parent" til B. Variablene repræsenter hændelser med et antal tilstande. En variabel kan f.eks. være en cykels farve, med mulige tilstande f.eks. blå,rød, gul osv. I et kausalt netværk repræsenterer en variabel et sæt af tilstande. En variabel er i netop *en* tilstand, som kan være ukendt af os.

10.2.1 Eksempel: For sent til middag

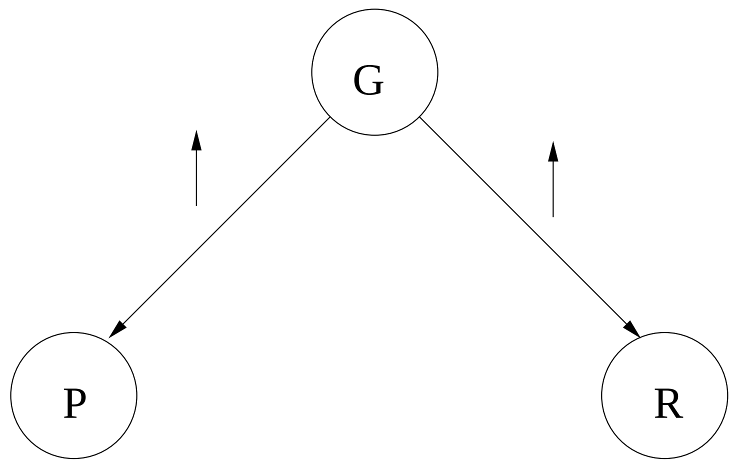
Fru Jensen, venter sin bror og søster, Rikke og Peter, til middag. De har aftalt at mødes kl 18.00 og da klokken bliver 18.15 begynder Fru Jensen at bekymre sig over hvor de bliver af. Hun spekulerer på om vejene kan være glatte, og hun ved at de begge er meget dårlige til at cykle, så hvis der er glatte veje kan de begge køre galt.

Hr. Jensen kommer ind, han siger at Peter lige er kørt galt. Peter? Hmm det kunne være værre, hvis der er glat så er Rikke garanteret også kørt galt, siger Fru Jensen. Jeg må hellere sætte maden til side. "Glat" siger Hr. Jensen, "Nej det er lang tid siden at det holdt op med at regne, og vejene er helt bestemt tørre nu". Fru Jensen er betrygget og tænker "Synd for Peter, lad os nu give Rikke 10 minutter mere".

For at formalisere historien, kan man lade hændelserne være repræsenterede af variable med to tilstande, enten *Ja* eller *Nej*. Hver hændelse er associeret med en sandsynlighed, dette giver tre variable:

1. Glatte Veje (G)
2. Rikke kører galt (R)
3. Peter kører galt (P)

G har den effekt at den forøger sandsynligheden for både R og P. Vi kan anse hændelsen G som en forøgende funktion fra sandsynligheden af årsagen til sandsynligheden af effekten [29]. Dette er illustreret i 10.1.



Figur 10.1: En netværks model af For sent til middag. De kausale pile forbundet fra årsagen G til hændelserne P og R og de små pile forbundet til sammenkædningerne indikerer retningen af årsagen på hændelsen.

Da Fru Jensen fik at vide at Peter var kørt galt, gjorde hun et ræsonnement i den modsatte retning af de kausale pile. Siden at årsagsfunktionen pegende mod P er blevet forøget, betyder det at den modsatte, pegende mod R, også er blevet forøget. Således får vi en forøgelse af sandsynligheden for G. Den forøgede tro på G, skaber dertil en ny antagelse, nemlig en forøget sandsynlighed for R.
Derefter, da hendes mand siger at vejene ikke kan være glatte, kan Peters uheld ikke påvirke hans forventning om vejenes tilstand, og som konsekvens har Peters uheld ingen indflydelse på R.
Dette er et eksempel på hvordan uafhængigheder og afhængigheder ændrer sig med den information, der er til rådighed. Når ingenting er kendt om vejenes tilstand, er R og P afhængige af hinanden, altså ændrer information om den ene altså også den anden. Hvorimod hvis vejenes tilstand er kendt, så er P og R pludselig uafhængige af hinanden, altså ændrer information om den ene ikke sandsynligheden for den anden. Dette kaldes også betinget uafhængighed. [29]

10.2.2 Ræsonnementer

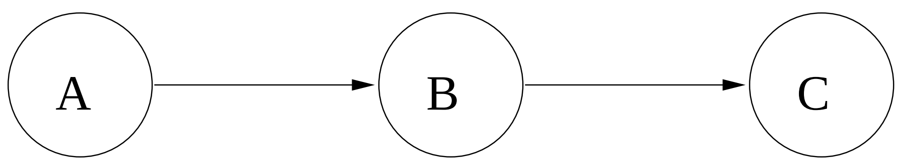
Modeller er en vejledning for måder at ræsonnerer omkring ukendte hændelser. Ræsonnementer i retning med de kausale pile er udtrykt:

Hændelsen A, forårsager hændelsen B med sandsynligheden x

Fra dette kan vi ræsonnere at hvis vi ved at A er sket, så er B også sket med sandsynlighed x.
Ræsonnementer i den modsatte retning af de kausale pile er derimod lidt mere ømfindigt. Indtil nu har vi kun sagt at sandsynligheden af årsagen A forøges når konsekvensen B er sket. Hvis man vil have en kvantitativ erklæring, er man nødt til at have en måde at invertere de kausale erklæringer på. Dette kommer vi ikke længere ind på. Yderligere information kan findes i [29] 2.4 og Kapitel 6.

Serielle forbindelser

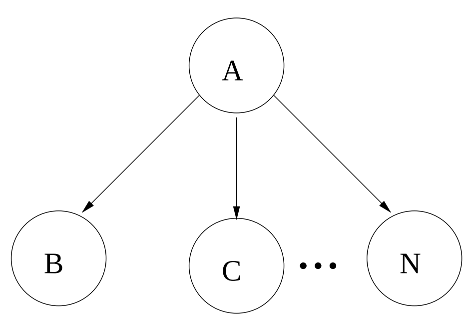
Vi betragter 10.2. A har indflydelse på B som også har indflydelse på C. Evidens på A vil påvirke sandsynligheden for B, hvilket derefter påvirker C. Ligeledes vil et bevis på C påvirke sandsynligheden for B og derigennem A. Hvis tilstanden af B er kendt, så er kanalen blokeret og A og C bliver uafhængige. Vi bruger notationen at A og C er separerede givet B, og når en tilstand af en variabel er kendt siger vi at den er *instantieret*. [29]



Figur 10.2: Seriel forbindelse. Når B bliver *instantieret* blokerer den for kommunikation mellem A og C

Afvigende forbindelser

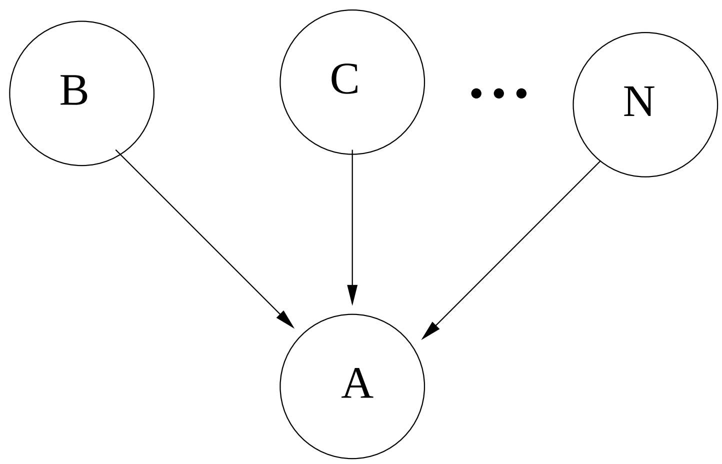
Situationen i 10.3 er en generalisering af eksemplet 10.2.1 *For sent til middag.*Enhver påvirkning kan passere mellem alle “children” af A medmindre at A’s tilstand er kendt. Vi siger at $B, C \ldots, N$ er separerede givet A. Så beviset kan blive transmitteret gennem en afvigende forbindelse. [29]



Figur 10.3: Afvigende forbindelse, Hvis A er instantieret, så blokerer den for kommunikation mellem alle dens “children”.

Konvergerende forbindelser

En beskrivelse af situationen i 10.4 kræver lidt mere forsigtighed. Hvis intet er kendt om A, undtagen hvad der kan udledes af fra dens *parents* $B, C, \ldots, N$, så er dens *parents* uafhængige: evidens på en af dem har ikke indflydelse på sandsynligheden for de andre.[29]
Hvis hvilken som helst form for evidens påvirker sandsynligheden for A, så er *parents* blevet afhængige. Evidenset kan orientere evidens på A eller det kan også være evidens fra et *child*. Konklusionen er at evidens kun kan overføres gennem en konvergerende forbindelse, hvis en hvilken som helst af variablene i forbindelsen, eller en af dens afkom, har modtaget evidens.



Figur 10.4: Konvergerende forbindelser. Hvis A ændrer sandsynlighed, så åbner den kommunikation mellem dens *parents*

10.3 Bayesianske Netværk

Et bayesiansk netværk er en model af et problem domæne, der repræsenterer kausale relationer mellem domæne variabler, og bliver brugt til at udregne sandsynligheds fordelinger af uobserverede variable givet de observerede variable. Relationerne mellem variablerne er givet som betingede sandsynligheder. Således består et Bayesiansk netværk af to dele: (1) en kvalitativ del og (2) en kvantitativ del. Den kvalitative del er givet uafhængigt af den kvantitative del, og indkoder de kausale relationer i domænets variable i form af en ikke cyklisk, orienteret graf, også kaldt DAG (Directed Acyclic Graph).
Den kvantitative del kaldes også for styrke. Dette bliver udtrykt ved at knytte tal til forbindelserne.
Lad A være en *parent* til B. Ved brug af kalkulus ville det være naturligt at lade $P(B|A)$ være forbindelsens styrke. Hvorimod hvis C også er en *parent* til A, som

illustreret i **??**, så giver de to konditionelle sandsynligheder $P(B|A)$ og $P(B|C)$ ikke alene noget hint til hvordan årsagerne fra A til B påvirker hinanden. Derfor må vi specificere $P(B|A, C)$.

10.3.1 Betinget Sandsynlighed

Betingede sandsynligheder er en af hjørnestenene i Bayesianske behandling af sandsynligheder. Hvis $P(A)$ af en hændelse A er givet, så er den givet afhængig af andre kendte faktorer. Enhver konstantering omkring sandsynlighed er en konstantering betinget af hvad der ellers er kendt.

Givet en hændelse B, er sandsynligheden for hændelsen $A = x$.

Dette skrives $P(A|B) = x$. Det skal dog nævnes at førnævnte ikke betyder, at når B er sand så er x sandsynligheden for A. Det betyder at hvis B er sand og alt andet kendt er irrelevant, så er $P(A) = x$.
Altså er:

$P(A) = P(A|B)$

10.3.2 Et eksempel:

Vi laver et eksperiment med tre mønter: en normal mønt, en deform som vil give krone og en deform som vil give plat. Vores møntkaster tager den ærlige mønt og ud fra udfaldet vil han således kaste en af de deforme mønter.

Figur 10.5: Senariet kan modelleres således. Evidens på E_2 giver forhøjet tro på hvilket udfald der var i E_1.

Sandsynligheden for at han i første kast med mønt 1 får plat eller krone må være: $P(Krone) = 0.5 = P(Plat)$. Sandsynligheden for at den deforme mønt der giver krone, mønt 2, tilskriver vi $P(Krone) = 0.9$ og $P(Plat) = 0.1$ og ligeledes omvendt for den mønt der plat, mønt 3.
Vi indfører betegnelserne E_1 (Kast med mønt 1) og E_2(Kast med mønt 2 eller mønt 3). Altså $P(E_1 = Plat) = 0.5$
Sandsynlighedsfordelingen for $P(E_1, E_2)$ ser således ud:

	E_1	Plat	Krone
E_2	-	-	-
Plat	-	0.45	0.05
Krone	-	0.05	0.45

Hvor sandsynligheden for f.eks $P(E_1 = Krone, E_2 = Krone)$ regnes ud i 10.9 med den generelle formel:

$$P(A,B) = P(A) * P(B) \tag{10.8}$$

$$\begin{aligned} P(E_1 = Krone, E_2 = Krone) &= \quad (10.9) \\ &= P(E_1 = Krone) * P(E_2 = Krone | E_1 = Krone) \\ &= P(E_1 = Krone) * P(mnt2 = Krone) \\ &= 0.5 * 0.9 \\ &= 0.45 \end{aligned}$$

10.4 Hvad kan vi bruge Bayesianske Netværk til?

Bayesianske netværk kan i visse tilfælde bruges til støtte for beslutningstagning.

10.4.1 Hvornår er det en god ide at bruge Bayesianske Netværk?

- Når der er en veletableret kendskab til den kausale struktur i et domæne
- Når dele af de kausale relationer er af stokastisk natur
- Når den kausale struktur hænger sammen som en kæde
- Når hændelserne i et domæne er veldefinerede
- Når sættet af alle mulige sager skaber en lukket verden af en håndterbar størrelse
- Når opgaven er terapeutisk(f.eks. ved at observere symptomer og risiko faktorer kan man estimere sandsynlighederne af de mulige årsager, og derudfra kan man beslutte sig til en behandlingsmetode)

10.4.2 Hvor skal man holde sig fra at bruge BN?

- Når hændelserne ikke er veldefinerede
- Når den kausale struktur er ukendt eller uinterresant (Det kræver altid en vis mængde data, at lære de kausale strukturer)
- Når den kausale struktur er deterministisk, når alm logik er tilstrækkelig

Kapitel 11

Vores egentlige løsningsforslag

11.1 Indledning

Vi har valgt at udvikle et spamfilter som vores løsningsforslag. Vi kan og vil ikke accepterer spam, vi vil ud fra dette faktum, bestræbe os på at bidrage med et konstruktivt kørende program. Vi har i forbindelse med Fagetsinformations teknologi valgt at implementere det i SML. Vi vil i det følgende afsnit forklare hvad vores intensioner er.

11.2 Flowchart

Se appendix 13.1.
Vores vision til det bedste spamfilter, er lavet ud fra de kriterier der er: (1) filteret skal være klientbaseret og er ikke beregnet til brug på mailserver (2) det er brugeren der i opbygger en database ved at bestemme om en mail er eller ikke er spam. På denne måde bliver det et meget personligt og yderste effektivt filter. [8]

11.2.1 Vores filtervision

Vi har lavet et flowchart over vores visionsfilter. og det vil vi give en forklaring på her:

Trin 1

Vores bruger modtager en e-mail..

Trin 2

Det første der sker her er at der bliver søgt efter afsenderen i e-mailens header. Når denne er fundet, benytter vi os af en sporingsalgoritme, der finder ud af om afsenderen eksistere. Gør afsenderen dette bliver headeren kontrolleret for fejl. Man kan forestille sig at en afsender udgiver sig for at være en anden end han egentlig er og har taget en andens adresse. Her kunne en fejl kan f.eks. være at IP adressen ikke er mellem 0-255. Det er de samme fejl som der bliver søgt efter i SpamAssassin og som står beskrevet nederst på side 116. [30]. Hvis ikke afsenderen bliver fundet bliver mailen slettet. Til sidst i dette trin søger vi efter en afsender reply i mailen, det gør vi på samme måde som vi søger efter afsenderen, med en sporingsalgoritme, som giver os besked på om der er en reel modtager. Hvis modtageren ikke eksisterer, eller der er andre dele af mail headeren der er forkert manipuleret, er der stor sandsynlighed for at mailen er en spam og den vil blive slettet, ellers bliver den sendt til trin 3.

Trin 3

Her bliver afsenderen først kontrolleret om denne står på brugens whitelist. Hvis afsenderen er på whitelisten bliver e-mailen sendt til trin 6 ellers bliver der kontrolleret om afsenderen står på brugrens egen blacklist, det vil sige om brugeren før har modtaget en mail fra afsenderen som brugeren selv har klassificeret som spam. Dette gør brugeren ved at bruge funktionen "blacklist" (trin 15). Hvis afsenderen før har været blacklistet bliver e-mailen sendt til trin 5, ellers bliver e-mailen sendt til trin 4.

Trin 4

Står afsenderen ikke på en af disse to lister bliver afsenderen kontrolleret om vedkommende står på en opdateret blacklist på Internettet. Bliver afsenderen fundet i online blacklisten, så bliver e-mailen sendet til trin 5, ellers bliver afsenderen sendt til trin 7.

Trin 5

Når en e-mail bliver sendt hertil, bliver den sendt til trin 14.

Trin 6

Hvis afsenderen er på brugens whitelist bliver e-mailen sendt til brugerens inbox (trin 13).

Trin 7

Afsenderen er ikke kendt og der er derfor ikke noget match og derfor sender vi hele e-mailen til trin 8.

Trin 8

E-mailen bliver opdelt i tokens som f.eks sex, xxx, join, søde ovs. Og der laves en lang list over alle e-mailens tokens. Trin 8 kalder databasen (trin 9) for at hente sandsynlighedsværdierne, ved statistisk udregning, for alle e-mailens tokens, og her efter laves en ny list (trin 10).

Trin 9

Dette er databasen for tokens og deres sandsynlighedsværdi målt i procenter for om det er et spam ord. Hvis der er nogle nye tokens på listen bliver disse tildelt en syndsynlighedsværdi på 0,4. Det her at alle tokens bliver opdateret, hvis der er en større eller mindre sandsynlighed for at tokens er et spam ord.

Trin 10

Listen fra databasen (trin 8). er nu klar. Det er stadigvæk alle e-mailens tokens, der står i den rækkefølge som de stod i e-mailen listen, f.eks.

- sex.......0.93
- xxx.......0.99
- pusser....0.01
- skat......0.30

Derfor bliver denne liste sendt til trin 11.

Trin 11

List som nu er kommet fra trin 9. bliver opstillet på en linær måde som i kan se her:

- xxx.......0.99
- sex.......0.93
- join......0.91
- offer.....0.90
- her ligger den neutrale grænse på 0.5
- skat......0.30
- hund......0.25
- søde......0.10
- pusser....0.01

Vi skal kun bruge en liste på 15 tokens ud af alle mailens tokens til at beregne sandsynligehed for om det er spam, og det gøre vi ved at bruge den "neutrale grænse" på 0.5. Den tager de tokens der ligger længst væk fra grænsen, altså dem nærmest 0.99 og ved 0.01. På denne måde får vi tokens med en høj og lav værdi til at brug til vores udregning af sandsynligheden for om det er eller ikke er spammail. De 15 tokens bliver som liste sendt til trin 12.

Trin 12

Den samlede sandsynlighed bliver udregnet, og efter den procentvise sandsynlighed som brugeren selv har sat. Derefter bliver mailen klassificeret som en spammail eller bare en mail. Spammails bliver sendt til trin 14 og e-mails som har en lavere sandsynlighed end den brugerbestemte, bliver sendt trin 13.

Trin 13

Brugerens inbox er der hvor mailen bliver lagt efter at have været igennem vores filter og ikke er blevet klassificeret som spam af filteret.

Trin 14

Dette er brugerens outbox hvor mailen bliver slettet. Vi har valgt ikke at returnere mailen for ikke at skabe mere unødigt trafik på Internettet, og ikke at medvirke til at spamme spammerne.

Trin 15

Vi har givet brugeren mulighed for at selv at kunne blacklistet afsendere som brugeren ikke vil modtaget e-mails fra. Dette har vi tænkt os at gøre ved at lave en blacklistnings knap til brugeren, som sender afsenderen til trin 18.

Trin 16

Når vi har givet brugeren en blacklistning knap, må vi selvfølgelig også give brugeren en whitelistnings knap, hvis brugeren ombestemmer sig og gerne vil have mail fra en afsender han tidliger har blacklistet. Og til alle dem brugeren gerne vil modtage mail fra lige meget hvad de indeholder f.eks. brugerens venner eller kollegaer. Når brugeren har brugt denne funktion bliver afsenderen sendt til trin 18.

Trin 17

Vi givet også brugeren en knap, så at han kan tilføje en mail som spam til databasen og mailens tokens bliver sendt til trin 19. som en tokenslist.

Trin 18

Hvis brugeren har brugt blacklist knappen eller whitelist knappen så vil den omtalte liste blive opdateret her. Hvis brugeren lige har whitelistet en afsender der stod i brugerens blacklist så vil afsenderen blive slettet fra blacklisten og omvendt, hvis bruger lige har blacklistet en afsender fra brugerens whitelist så bliver afsenderen blacklistet.

Trin 19

Når brugeren har brugt funktions knappen "tilføj spam" bliver databasen opdateret og de tokens der er i tokenslist fra e-mailen ,som blev klassificeret som spam, af brugeren vil får en højere sandsynlighedsværdi end før.

Trin 20

Vi kunne f.eks. også have valg at når en mail blev whitelistet, så vil databasen blive opdateret sådan at sandsynlighedsværdien blev mindre for de tokens der var i mailen. eller omvendt hvis en mail blev blacklistet at databasen blev opdateret sådan at sandsynlighedsværdien blev større for de tokens fra denne mail.

11.3 Beskrivelse af de enkelte moduler i sml filtret

11.3.1 Modul 1, tokenizer

Vores spamfilter skal kunne tage en hvilken som helst fil, som vi ønsker skal filtreres, opdele teksten via en tokenizer, altså en funktion der deler teksten op i tokens, hvorved vi får et output i form af en liste, hvor alle de enkelte ord står efter hinanden adskilt med et komma. Et eksempel kunne være:

```
"jeg glæder mig til sommer"
[‘‘jeg’’,’’glæder’’,’’mig’’,’’til’’,’’sommer’’]
```

```
val DBfile="lsb.dbx";
```

Her bliver DBfile som er forkortelsen for Database file tildelt værdien "lsb.dbx"og enhver fremtidig reference til DBfile går tilbage hertil.
Herefter bliver der startet en funktion med forkortelsen fun, funktionens navn er iMtt(in Mail to tokens), og (fileIn) er en variabel der kan indsættes.

```
fun iMtt(fileIn) =
     let
         val file = TextIO.openIn fileIn
```

Filen fileIn refererer til navnet file, Herefter opretter TextIO.openIn en inputstrøm af fileIn, med andre ord, filen bliver proppet ind i ram lageret.

```
val s = TextIO.inputAll file
```

Filen file får nu navnet s, herefter bliver filen indlæst fra en inputstrøm, dvs. filen bliver læst.

```
in
     String.tokens Char.isSpace s
end
```

her bliver filens tekst til en liste af tokens, ved kald af den tidligere reference *s*. Det vigtige her er at notere at funktionen iMtt altså opdeler med *Char.isSpace* altså der hvor der er mellemrum.

11.3.2 Modul 2, Database

specifikation:
Input: En *string list*, denne liste indeholder alle de ord der findes i den pågældende mail.
Output: En *real list*, som indeholder sandsynlighederne for hvert ord i den liste af ord some var input.
Beskrivelse af løsning:

```
fun STPlist (stringlist)=
    let
        val db = DataBase(DBfile)
    in
        Plist(stringlist,db)
    end;
```

STPlist(Stringlist To Probability list), kaldes med en *string list* som parameter. Denne funktion har til opgave ud fra den givne liste af ord at opbygge en liste af hvert af disse ords “spamsandsynlighed”. For at kunne gøre dette opbygger den en *tokenList*, en erklæret type som er en liste af records der indeholder alle ord fra vores database fil. Denne *tokenList* skal se således ud:

```
type token = {ts: string, s: real, n: real};
type tokenList = token list;

ex.: [{"ost",2.0,4.0},{"fisk",34.0,2.0},..........]
```

Hvor ts = token i form a en string, s = antal gange set i en spam mail, n = antal gange set i en ikke spam mail. For at kunne få opbygget en *TokenList* kaldes funktionen DataBase med parameteren file, som er navnet på den fil der indeholder vores database. Database kaldes i vores tilfælde med DBfile(se STPlist).

```
fun DataBase (file) = ReadFile(TextIO.openIn file);
```

DataBase funktionen kalder ReadFile for at læse i database filen

```
fun ReadFile file =
      if TextIO.endOfStream file
      then (TextIO.closeIn file; [])
      else (ReadLine file)::(ReadFile file);
```

I ReadFile laves et check på om TextIO.endOfStream er true, hvis det er tilfældet så er det fordi, der ikke er mere data i input filen, så skal funktionen terminere. Hvis det ikke er tilfældet laver den først et kald til funktionen ReadLine hvorefter den laver et rekursivt kald til sig selv, og dermed opbygger en liste af outputs fra ReadLine funktionen. (indikeret ved de to ::(kaldet kons))

```
fun ReadLine file =
      let
        val line = TextIO.inputLine file
        val lineList = String.tokens Char.isSpace (line)
      in
        TokenRecordBuilder lineList
      end;
```

ReadLine læser en linje fra database filen og laver en *string list* af de informationer, der findes i den pågældende linje i filen(læg mærke til at Char.isSpace her betyder at filen skal være opbygget så informationerne er adskilt med et mellemrum).

```
ex.
        filen indeholder:
        ost 23.0 3.0
        fisk 30.0 5.0

lineList = ["ost","23.0","3.0"]
```

Herefter kaldes TokenRecordBuilder med listen af de string tokens som lige er blevet læst i filen, den skal sørge for at opbygge en record af disse tokens.

```
exception FileError;

fun TokenRecordBuilder [s,x,y] =
      {ts = s, s = valOf (Real.fromString x),
               n = valOf (Real.fromString y)}

  | TokenRecordBuilder _ = raise FileError;
```

ValOf funktionen sørger for at x og y bliver lavet om til deres eksakte værdier så de ikke har typen string men typen real. Hvis vi har et illegalt input kalder funktionen FileError execptionen (denne bliver dog ikke fanget, men kunne udbygges til at gøre det).
Nu er der blevet bygget en *tokenList*. (se evt ReadFile funktionen, her laves en liste af outputs fra ReadLine, som er records af typen *token*)

```
fun STPlist (stringlist)=
    let
        val db = DataBase(DBfile)
    in
        Plist(stringlist,db)
    end;
```

Variablen db er nu en *tokenList*, så kaldes Plist med parameterne stringlist og db som skal opbygge en liste over spam-sandsynlighederne.

```
fun Plist ([],db:tokenList) = []
  | Plist (s::lA,db) = findTokenP(s,db)::Plist(lA,db);
```

Sandsynlighedslisten laves ved hjælp af en rekursiv funktion, funktionen kalder findTokenP (find Token Probability) på den første token i stringlisten og derefter kalder sig selv med en liste der nu er en token mindre, indtil den er nået hele stringlisten igennem.

```
fun findTokenP (_, []: tokenList) = 0.4
  | findTokenP (t, {ts,s,n}::tList) =
    if t=ts then s/(n+s)
    else findTokenP(t, tList);
```

Funktionen findTokenP finder først den pågældende token t, i *tokenList*-repræsentationen af databasen v.h.a et rekursivt kald hvor den løber hele listen igennem indtil den finder t. Når t er fundet i listen udregnes sandsynligheden ved denne formel: s/(n+s), s = antal gange set i spam mail, n= antal gange set i ikke spam mail. Outputtet bliver så tilsidst en liste af sandsynligheder.

11.3.3 Modul 3, Sandsynlighedsregning

Vi har valgt at bruge Paul Grahams algoritme til at udregne den samlede sandsynlighed for om en e-mail er spam [8]. For sammenligning har vi prøvet at implementere en naiv bayesiansk metode, for mere nøjagtigt at kunne bestemme sandsynligheden for, at en given e-mail er en spam mail. Der findes en god tilgang hertil på [5]. (se afsnittet naiv bayesiansk tilgang til spam) Forklaring på Paul Grahams algoritme kan findes på [21] Denne sandsynlighed, kræver at alle faktorer er uafhængige, dvs at det ikke er et bayesiansk netværk.
Vi har valgt at beregne en e-mails samlede spamsansynlighed udfra de tokens spam sansynligheder vi lige har beregnet. Paul Graham konstanterer at følgende formel vil udregne den samlede sansynlighed for at e-mailen er spam, alle tokens taget i betragtning:

$$P(Mail = Spam) = \frac{a * b * c * \ldots n}{(a * b * c * \ldots n) + ((1-a)(1-b)(1-c)\ldots(1-n))} \tag{11.1}$$

Vi går her udfra at “a” og “b” er 2 stykker uafhængige tokens med hver deres egen sansynlighed for spam eller ej. Vi har i denne linie kode, lavet den selvsamme formel, hvor vi udregnet den samlede sansynlighed for spam:

```
fun sA [] = 1.0
| sA (x::l) = x * sA(l);

fun sB [] = 1.0
| sB (x::l) = (1.0-x)*sB(l);

fun ss (listIn) = sA(listIn)/(sA(listIn)+sB(listIn));
```

Funktionen sA udregner et samlet produkt for atributterne $a, b, c, d, e \ldots fn$ og sB udregner et samlet produkt for de modsatte atributter $(1-a)(1-b)\ldots(1-n)$. Funktionen ss kalder de to førnævnte for at beregne den samlede sandsynlighed, vha. Paul Grahams metode.
Når vi så møder et ord, som vi aldrig har set før, tager vi Paul Grahams erfaringer til os. Da han via forsøg har fundet at en god værdi for ukendte ord er 0.4 da vi herved giver en god score, og stadig vægter non-spam værdi højere. Vi gør naturligvis dette for at undgå falske posetive, hvilket er ”korrekte” e-mails som bliver fanget i filtret ved uheld. Princippet i denne udregning forløber sig så senere sådan at de værdier vi udregner efter bliver indstillet løbende ved at man kører en enorm mængde e-mails igennem filtret og specificerer om det er spam eller ej. Værdierne skal så for de forskellige token hæves og sænkes. Derved vil man efter en vis mængde e-mails, både spam og non-spam, få nogle værdier som gerne skulle give mindst mulige falske posetive og fange langt de fleste spam e-mails. Det vi altså gør er at tage forskellige stykker, uafhængige, beviser og beregner en samlet sansynlighed for om de udgør spam eller ej.

Et eksempel Paul Graham selv bruger er således:
Hvis en person er over 1.90m høj er sansynligheden for at han spiller basked X.
Hvis personen bærer en basketball er sansynligheden Y.
Hvis personen både er over 1.90m høj og bærer en basketball bliver den samlede sandsynlighed for at han er basketball spiller større, end de 2 stykker beviser tilsammen. Ligeledes ville andre beviser kunne sænke sandsynligheden, f.eks kunne fodboldstøvler sænke sandsynligheden for basketball betragteligt.
Hvis man så forestiller sig at vi ikke prøver at finde sandsynligheden for at en person er basketspiller, men i stedet sandsynligheden for en e-mail er spam, vil hver token som har en værdi enten tale for eller imod hele mailen som værende spam. F.eks. ville ordet ”sex” nok tale for, og ordet ”biograf” tale imod. Ligeledes vil alle småord, vægte imod spam. Så mails som indeholder ”sex” og dog ikke er spam vil i de fleste tilfælde blive lukket igennem, da ”sex” tæller så og så meget op, mens de andre ord nok ville trække sandsynligheden ned igen.

11.3.4 Modul 4, Bruger indstilling af spamvalue

Funktionen *inputspamvalue* er en lille funktion, der henvender sig til brugeren. Den tager en fil ind og giver et output til en fil igen.

```
open TextIO;

fun inputspamvalue (fileOut)=
    let
        val definer =
            (output(stdOut,"Definer din spam graense i
                  intervallet [0;1] med 2 decimaler\n ");
            flushOut(stdOut));
        val os = openOut fileOut
        val sv = inputN(stdIn, 4 );
in
        output (os, sv);
        flushOut(os)
end;
```

Den åbner først *TextIO* pakken, primært for at aflaste programmøren for at skrive *TextIO* foran kommandoer som *TextIO.openOut*, der sålede bare bliver til *openOut*.
Val definer... giver et output til brugeren om at indtaste hans personlige grænse for hvornår en email er spam. Dette bliver skrevet på skærmen med *flushOut*, hvorefter *inputN(stdIn, 4)* fanger inputtet. *output(os,sv)* fører inputtet over til den stream der er åbnet i *openOut fileOut* og *flushOut(os)* skriver steamen til filen.

11.3.5 Modul 5, Opdaterer

specifikation:
Input: En *string* i form af navnet på en fil(email).
Output: Dette modul overskriver den eksisterende database fil med en ny opdateret fil.

```
fun addSpam(mailFile) =
    let
        val mail = iMtt(mailFile)
        val db = DataBase(DBfile)
    in
        writeToDisk(DBupdate (1.0,0.0,mail,db),DBfile)
    end

fun addHam(mailFile) =
    let
        val mail = iMtt(mailFile)
        val db = DataBase(DBfile)
    in
        writeToDisk(DBupdate (0.0,1.0,mail,db),DBfile)
    end
```

Funktionerne *addHam* og *addSpam* bruges til at opdatere databasen udfra en given tekst fil. For at kunne opdatere bliver der først genereret to variabler(mail,db), disse indeholder henholdsvis en *string list* og en *tokenList* (se modul 1 og 2 for hvordan *iMtt* og *DataBase* funktionerne virker).
Før funktionerne skriver den nye fil, opdateres databasen med funktionen *DBupdate*.

```
fun DBupdate (spam,nospam,[],db )= db
  | DBupdate (spam,nospam,x::l,db) =
    let
        val temp = updateToken(spam,nospam,x,db)
    in
        DBupdate(spam,nospam,l,temp)
    end;
```

DBupdate tager 4 argumenter, spam og nospam er real værdier som skal være 1.0 eller 0.0. Den næste skal være en *string list*(listen af ord fra emailen som skal opdatere databasen) og den sidste en *tokenList*(databasen som den ser ud før vi begynder at opdatere den).
Temp variablen sørger for at opdatere den første token i stringlisten fra mailen, så *temp* nu indeholder den foreløbigt opdaterede database. Med et rekursivt kald

løber funktionen så hele stringlisten igennem indtil den er tom. Når funktionen terminere står den med en ny opdateret database i form af en tokenList.
Hver enkelt token bliver opdateret således:

```
fun updateToken  (spam,nospam,t, []: tokenList) =
                 [{ts=t,s=2.0,n=3.0}]
 |  updateToken (spam,nospam,t, {ts,s,n}::tList) =

    if t=ts then {ts=ts, s=s+spam,n=n+nospam}::tList
    else {ts=ts,s=s,n=n}::updateToken(spam,nospam,t, tList);
```

UpdateToken opdatere den pågældende token i databasen som bliver givet. Den laver et linært check, hvor den kører hele *tokenList*en igennem indtil den finder den token *t*, som er givet med. Når den har fundet denne opdaterer funktionen listen og giver nu en ny liste tilbage til *DBupdate*. Den opdaterede liste laves udfra et rekursivt kald, således at funktionen bliver ved med at kalde sig selv på resten af databasen indtil den har fundet *t*. Hvis *t* ikke findes, dvs. den har kørt hele databasen igennem og står nu med en tom liste, oprettes *t* nu som en ny token i databasen. Udfra Paul Grahams [9] teori oprettes manglende ord i databasen med en begynder værdi på 0.4. [ts=t,s=2.0,n=3.0] så s/(n+s) = 0.4.

```
    ...
         writeToDisk(DBupdate (1.0,0.0,mail,db),DBfile)
end;
```

DBupdate har nu sørget for at databasen er blevet opdateret med ordene fra mailen og er klar til at blive skrevet ud i en fil.

```
fun writeToDisk (tList,file) =
    let
        val opf = TextIO.openOut(file)
        val stringlist = ush(tList)
    in
        TextIO.flushOut(writeToStream(stringlist,opf))
    end;
```

Funktionens input er den opdaterede liste samt navnet på filen i string form der skal skrives til. For at kunne skrive i en fil laves den opdaterede database liste af typen *tokenList* om til en *string list*. Dette gøres ved at kalde *ush* funktionen med database listen som input.

```
fun ush ([]:tokenList) = []
  | ush ({ts,s,n}::tList) = String.concat([ts," ",Real.toString(s)," ",
         Real.toString(n)])::ush(tList);
```

For kunne skrive dem rigtigt ud i filen konstanteres hver record i *tokenList*en til en streng. Resten af string listen laves ved et rekursivt kald.

Listen af strenge, hvor hvert element repræsenterer en token med de relevante informationer om antal spam gange optrådt i spam/nospam mails, skrives nu til bufferen(output stream) v.h.a funktionen *writeToStream*.

```
fun writeToStream ([],stream) = stream
 |  writeToStream (x::l,stream) =
    let
        val newstream = stream
    in
        TextIO.output(newstream,x);
        TextIO.output(newstream,"\n");
        writeToStream(l,newstream)
    end;
```

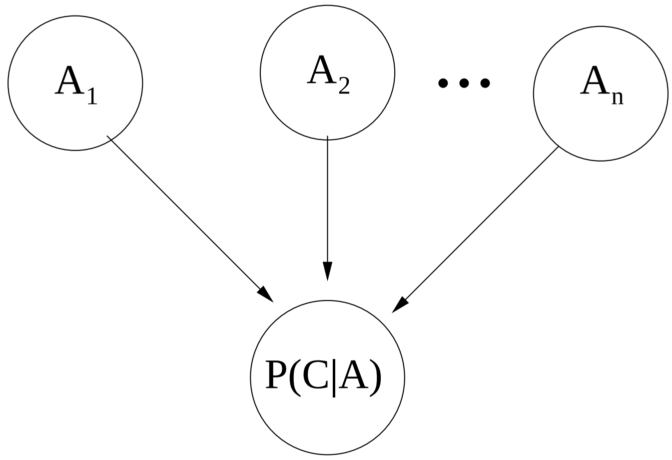
Her udskrives hvert element i input listen til output streamen(""tages med for at skrive på en ny linje). Denne stream opbygges ved rekursivt kald, og ender med en stor stream som udskrives til en fil i *writeToDisk*.

```
        TextIO.flushOut(writeToStream(stringlist,opf))
end;
```

11.4 Bayesiansk tilgang til spam

Vi har hørt en del kritik omkring Paul Grahams algoritme for klassefisering af spam mails[17], denne kritik har ført til at vi har kigget på Bayesianske Netværk og prøvet at konstruere en metode til at udregne sandsynligheden for at en given mail er spam v.h.a. et naivt bayesiansk netværk.

Bayesiansk netværks metode:



Figur 11.1: Beregningsprocesen modelleret

Ifølge Bayes regel er

$$P(C = c|A_1 = a_1, A_2 = a_2..A_n = a_n) = \frac{P(A_1 = a_1, A_2 = a_2..A_n = a_n|C = c)}{P(A_1 = a_1, A_2 = a_2..A_n = a_n)} * P(C = c) \tag{11.2}$$

Denne sandsynlighed skal forklares som: P(C=c|A=a) er sandsynligheden for at C, i vores tilfælde $C = SPAM$, indtræffer hvis en given atribut A, i vores tilfælde ord fra en mail, er indtruffet. Altså at mailen er en spammail forudsat den indeholder de pågældende ord.
C kan have værdierne $\{SPAM, NOTSPAM\}$, da det vigtigste for os er at se om det er en spam mail, har vi valgt kun at kigge på $C = SPAM$
$A_1, A_2, ...A_n$ er tokens fra mailen, eks. er orderne "fisk", "sex", ..
$a_1, a_2, ...a_n$ er om en given token findes i den mail, som vi udregner sandsynligheden for, dvs 1 for at tokenen er der, eller 0 hvis den ikke findes i mailen.
For at kunne beregne den samlede sandsynlighed, har vi delt det op i 3 dele:

1. Udregning af $P(A_1 = a_1, A_2 = 1a_2..A_n = a_n|C = c)$
   Bayes regel anvendes, så udtrykket kommer til at se således ud:

   $$\begin{aligned} &P(A_1 = a_1, A_2 = 1a_2..A_n = a_n|C = c) = \\ &P(A_1 = a_1|A_2 = a_2, A_3 = a_3, ..A_n = a_n, C = c) * \\ &P(A_2 = a_2, A_3 = a_3, ..A_n = a_n|C = c) \end{aligned} \tag{11.3}$$

   For at få isoleret *Attribut 2* benytter vi Bayes regel endnu en gang på det sidste led:

$$
\begin{aligned}
& P(A_1 = a_1, A_2 = 1a_2..A_n = a_n | C = c) = \\
& P(A_1 = a_1 | A_2 = a_2, A_3 = a_3, ..A_n = a_n, C = c) * \\
& P(A_2 = a_2 | A_3 = a_3, ..A_n = a_n, C = c) * \\
& P(A_3 = a_3, ..A_n = a_n | C = c)
\end{aligned} \tag{11.4}
$$

Bayes regel kan nu benyttes til at isolere hvert led i udregningen indtil man har:

$$
\begin{aligned}
& P(A_1 = a_1, A_2 = 1a_2..A_n = a_n | C = c) = \\
& P(A_1 = a_1 | A_2 = a_2, A_3 = a_3, ..A_n = a_n, C = c) * \\
& P(A_2 = a_2 | A_3 = a_3, ..A_n = a_n, C = c) * \\
& P(A_3 = a_3 | A_4 = a_4, ..A_n = a_n | C = c) * ... * P(A_n = a_n | C = c)
\end{aligned} \tag{11.5}
$$

Nu antages, at alle atributterne er uafhængige, d.v.s.at vi formoder at sandsynligheden for en attribut ikke bliver påvirket af om der findes en anden given attribut i teksten. Altså at sandsynligheden for feks. "free"ikke afhænger af om vi har set ordet "offer"eller "dollar". Dette er ikke idealt, men en forsimplet udgave af verdnen, og dermed en naiv indgangs vinkel. Dette giver os at følgende er sandt:

$$
\begin{aligned}
& P(A_1 = a_1, A_2 = 1a_2..A_n = a_n | C = c) = \\
& P(A_1 = a_1 | C = c) * P(A_2 = a_2 | C = c) * \\
& P(A_3 = a_3 | C = c) * \ldots * P(A_n = a_n | C = c)
\end{aligned} \tag{11.6}
$$

Værdierne for de enkelte led kan approximeres ved at tælle i databasen, altså at tælle hvor mange gange, attributten har optrådt i forgående *spam* eller *nospam* mails.

$$
P(A_1 = a_1 | C = c) = \frac{count(A_1 = a_1, C = c)}{count(C = c)} \tag{11.7}
$$

2. Udregning af $P(C)$
   Dette er baggrunds sandsynligheden, denne udregnes fra data om hvor mange mails har været *spam* og hvor mange har været *nospam*.

$$
P(c) = \frac{count(C = c)}{count(C)} \tag{11.8}
$$

3. Udregning af $P(A_1 = a_1, A_2 = a_2, ..A_n = a_n)$
   Da vi har antaget at attributterne er inbyrdes uafhængige så ser det således ud:

$$
P(A_1 = a_1, A_2 = a_2, ..A_n = a_n) = P(A_1 = a_1) * P(A_2 = a_2) * ... * P(A_n = a_n) \tag{11.9}
$$

Hvilket også kan udregnes ved at tælle i databasen:

$$P(A_1 = a_1) = \frac{count(A_1 = a_1)}{count(C)} \tag{11.10}$$

For yderligere forklaring se eksempel i appendix 14.4
Vi har ikke fuldendt implementeringen af den naivt bayesianske sandsynligheds algoritme vi har derfor undladt at beskrive vores nuværende implementering af den bayesianske del.

11.5 Test af filter

For at teste det filter vi har programeret startede vi med at finde 18 emails, som skulle repræsentere vores spammails. Disse e-mails har vi taget udfra en e-postkasse, vi opretede på Hotmail.com tidligt i projektforløbet. 18 emails er langt fra nok til at teste vores filter ordentligt, men fordi vi ville holde det simpelt og til sidst var under tidsmæssigt pres, valgte vi at teste ud fra de få e-mails vi havde. Vi fandt derefter 18 e-mails som skulle repræsentere vores no spam e-mails. Disse kom fra en af vores egne e-mail adresser. Vi lavede så en kontrol gruppe på 6 e-mails som skulle se om vores filter havde nogen effekt.
18 emails blev lagt ind i databasen som spam e-mails. Vi checkede databasen og så, at der ikke forekom nogle fejl under inlæsningen af e-maillene. Denne del gik helt som forventet uden at vi opdagede nogen uregelmæssige tal i databasen.
18 emails blev lagt ind i databasen som no spam emails. Vi checkede igen databasen for at kunne fange eventuelle fejl, som kunne opstå i denne fase. Heller ikke denne gang blev vi overrasket, og alt gik som planlagt.
Herefter prøvede vi at køre alle 36 igennem for at se om filteret kunne identificere hver enkelt e-mail, som den gruppe den kom fra. Ud over en enkelt, der lavede en *div*(her gætter vi på, at der på et tidspunkt blev divideret med 0) fejl i udregningen, kunne filteret klasifisere hvert enkelt e-mail rigtigt. Dette var dog ikke helt uforventet, da databasen er lavet ud fra nettop disse e-mails.
Der var dog en overraskelse da vi kørte alle mails igennem, nemlig at sandsynlighederne, der blev udregnet var meget anderledes end forventet. Vi havde forventet tal imellem 0 og 1, men i stedet fik vi nogle ekstreme resultater. Den eneste der synes at holde sig inden for grænsen var Paul Grahams algoritme. den gav resultater meget tæt på 1 hvergang vi kørte en spammail igennem.
Vi kørte nu de 5 no spam e-mails igennem, som var i kontrolgruppen. Uden fejl karakteriserede filtret de 5 mails som no spam, men stadigvæk med meget extreme resultater i sandsynlighedsberegningen. 1 spammail blev nu kørt igennem for at teste, denne lignede meget de e-mails fra opbygningen af databasen, så ikke uforventet blev den klasificeret som spam.
Konklusionen på det hele blev, at vores kontrolgruppe og mængde af mails til opbygning af databasen, var alt for lille. Vi undrede os meget over resultaterne

og kunne ikke finde nogle årsager til at de skulle blive så extreme, og at den stadig kunne finde ud af at klasificere e-mails rigtigt. Ud fra denne test kunne vi ikke sige så meget om vores filters korrekthed.

11.6 Kritik af filter

Efter testen på filtret og arbejdet med at tilpasse algoritmerne i programeringssproget SML, har vi fundet nogle punkter, hvor vi kunne have udvidet og forbedret meget på det endelige pogram.

Søgnings algoritmer kunne væsentligt forbedres. Vi har i vores filter brugt rekursive funktioner, til at løbe lister linært igennem. Alle disse tidskrævende gennemløb, kunne have været meget forbedret hvis vi havde opretholdt en ordnet datastruktur i databasen. F.eks. kunne vi have valgt at ordne databasen i alfabetisk rækkefølge, hvilket ville kunne give en kortere søgningstid. Vi observerede allerede at programmet blev langsomere for hver e-mail, der blev lagt i databasen.

Vi har valgt ikke at gøre så meget ud af hvordan programmet udtager tokens fra filtret. Der bliver kun lavet nye tokens, hvis der er mellemrum imellem et ord. Bl.a. web adresser bliver ikke delt op i flere tokens, hvilke måske kunne være med til at lave en smartere opdeling af tokens og hjælpe med til filtrets succesrate.

Vi kendte ikke så meget til programeringssproget da vi startede programeringen af filtret, vi kunne måske have tænkt os en anderledes repræsentation af databasen frem for en liste af records. Det kunne tænkes, at man skulle arbejde med arrays eller andre former for datastrukturer, som ville være mere hensigtsmæssige at arbejde med.

Der er en fejl i filtret ved tildelingen af værdien til en token, der optræder for første gang, dette kommer sig af, at vi først satsede på at bruge Paul Grahams algoritme og ikke et bayesiansk netværk. Vi er klar over, at der er et problem, men har ikke formået at rette det.

Der er sikkert en del ting, der kunne forbedres i algoritmerne for at forbedre hastigheden og andre væsentlige punkter, men generelt er vi godt tilfredse med resultatet, som det ser ud lige nu. Vi er også sikre på, at hvis vi havde den nødvendige tid, kunne vi rette disse fejl og mangler.

Kapitel 12

Konklusion

Vi har gennem vores analyser af nuværende løsningsmodeller, herunder startende med lovgivning til tekniske løsninger, frembragt vores egen løsningsmodel. Vi fandt ud af at pt. er lovgivningen ikke tilstrækkelig til at stoppe spam, den problematiseres yderligere ved at landslove ikke vil kunne strække sig udover landets grænser, hvilket internettet gør. Nuværende bekæmpelse af spam i samfunds regi foregår hovedsageligt på det tekniske plan, hvor man prøver at stoppe spam ved hjælp af filtre, og informationer til at undgå spam. Vi har gennemgået de nuværende tekniske løsninger, og konluderet at det mest effektive, og alment mest brugte er spamfiltre, hvorefter vi har valgt at vores løsningsmodel skal være et sådan filter. Til udviklingen af vores filter har vi fået inspiration fra eksisterende filtre, og de kommende filtres teorier. Dette filter blev implementeret i SML, eftersom det er det programmeringssprog som vi fulgte gennem Fagets Informationsteknologi. Vi besluttede at bruge idéer grundlagt af Paul Graham [8], som er en af forkæmperne for at stoppe spam. Vi valgte derfor at opbygge vores eget spamfilter omkring hans metode. Vores egen løsningsmodel er absolut ikke optimal, men blot et eksempel på at metoderne virker i praksis. Filtret fungerer udelukkende på et eksperimentalt plan, og er ikke tiltænkt at implementeres i noget styresystem. Yderligere er vores filter stærkt ramt af at vores egne programmeringskundskaber absolut er begrænsede. Vi havde ikke helt de rigtige kundskaber for at kunne udvikle filtret til det fuldeendte, vi løb gentagende gange ind i tekniske problemer med SML, men overkom med rimelighed problemerne. Vores resultat er et fungerende filter.

Litteratur

[1] http://www.mailtraq.com/KB02052301.html, 10-11-02.

[2] Spam Abuse, *Spam abuse*, http://www.cauce.org.

[3] Jens Bertelsen, *Første danske spam-retssag*, Http://www.computerworld.dk/default.asp?mode=2ArticleID=16911, 17-11-02.

[4] CAUCE, *About cauce*, http://www.cauce.org.

[5] Charles Elkan, *Naive bayesian learning*.

[6] fabel, *fabel*, set d. 18-11-02.

[7] GAZnet, *Gaznet*, http://www.gaznet.au.com/spam/, 12-12-02.

[8] Paul Graham, *A plan for spam*, http://www.paulgraham.com/spam.html, 29-10-02.

[9] ______, *Spam is different*, http://www.paulgraham.com/dpamdiff.html, 10-11-02.

[10] David Heckerman, *A tutorial on learning bayesian networks*, 18-11-02.

[11] imc.org, *Thoose bulk email blues*, http://www.imc.org/bulk-email.html, 20-11-02.

[12] Jan Jacobsen, Krag, *Interview*, Hans ReitZels forlag.

[13] Hagen Jørgensen, *Ændring af markedsføringsloven - uanmodet henvendelse til bestemte aftagere*, http://www.fs.dk/jura/loveregl/mfl/$00_fouan.pdf, 17-11-02$.

[14] Microsoft, *Hotmail*, http://www.hotmail.com/, brugt til forsøg i gruppen, og set gentagende gange i hele forløbet.

[15] Wladyslaw Pietrazek, *Interview med en system adminstrator*, 21-11-02.

[16] Eric Raymond, *Bogofilter*, http://www.tuxedo.org/ esr/bogofilter/, 1-12-02.

[17] Gary Robinson, *Spamdetection*, http://radio.weblogs.com/0101454/stories /2002/09/16/spamDetection.html, 29-10-02.

[18] Simon Schwartz, Allan og Garfinkel, *Stopping spam*, O'Reilly Associates,Inc, 1998.

[19] Brad Templeton, *Origin of the term "spam"to mean net abuse*, www.templetons.com/brad/spamterm.html, 25-11-02.

[20] ukendt, *åben relays*, Http://www.ordb.org/, 08-10-02.

[21] ______, *Combining probabilities*, http://www.mathpages.com/home/kmath267.htm, set 15-12-02.

[22] ______, *Ghostsender*, Http://www.ltips.net/ghostsenderindex.html, 15-10-02.

[23] ______, *Internettets historie*, www.internettet.subnet.dk/historie.htm, 06-12-02.

[24] ______, *Lbk nr. 699 af 17/07/2000*.

[25] ______, *Macroware*, Http://www.computerworld.dk/default.asp?mode=2ArticleID=14648, 19-10-02.

[26] ______, *Spam fighter script*, http://www.website101.com/SpamFilter/spam$_{s}cript.html, 25-11-02$.

[27] ______, *Spamassassin*, http://spamassassin.taint.org/, 23-10-02.

[28] ______, *spamfilter*, Http://www.mailwasher.net/index.php, 26-10-02.

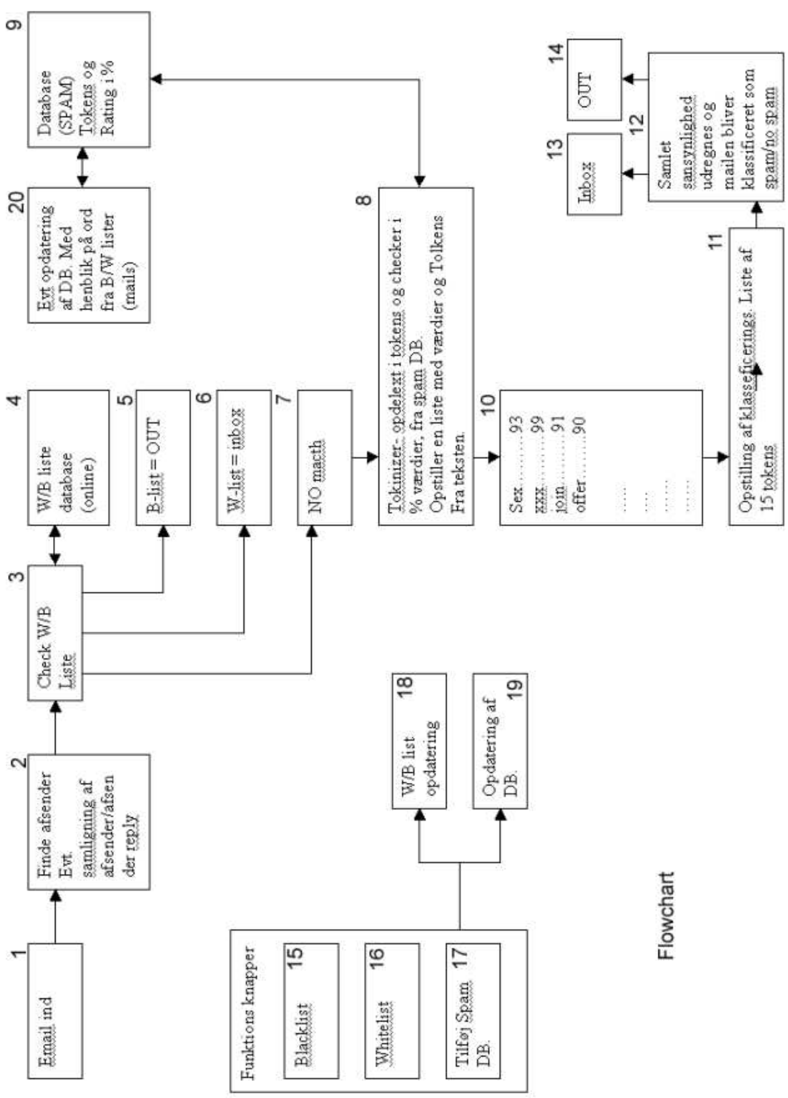
[29] Finn V. Jensen, *An introduction to bayesian networks*, UCL Press, 1996.

[30] Wallace Wang, *Internet - hacking, virus og faldgruber*, danish language edition published by ingeniøren a/s copyright 2000 ed., 1.udgave 1.oplag 2000.

Kapitel 13

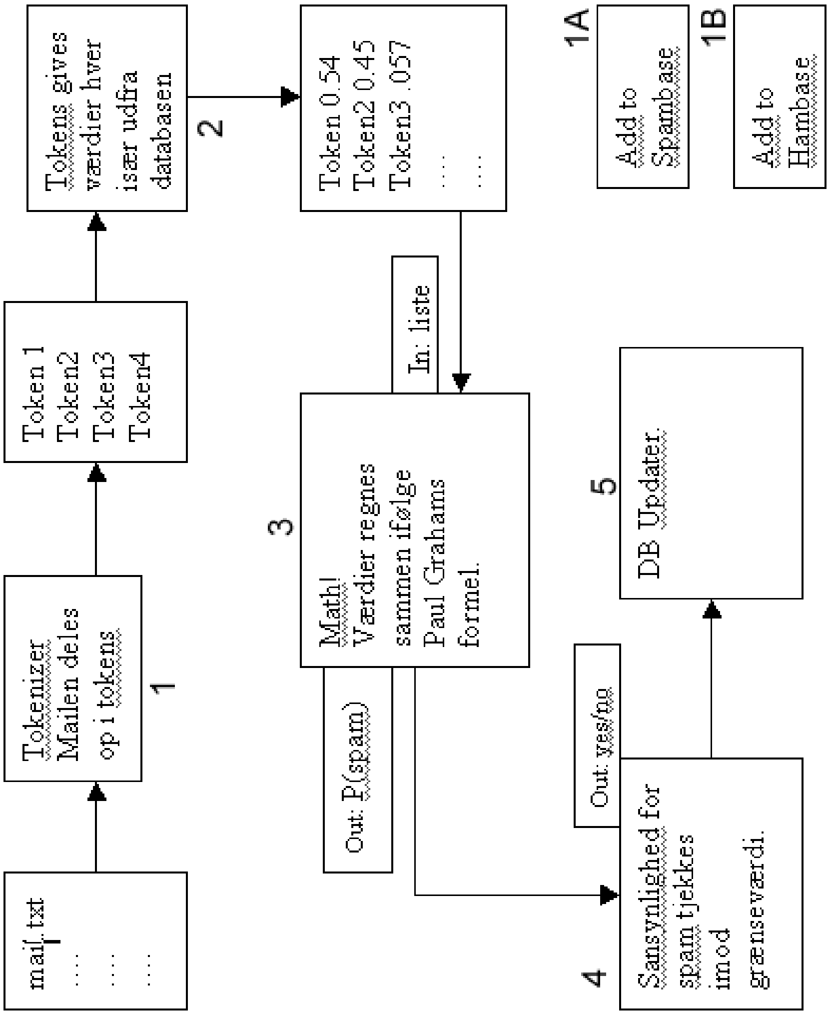
Appendix

13.1 Flowchart



Figur 13.1: Dette er vores flowchart over vores filter, med ialt 20 trin som der er beskrevet i afsnitet 11.2.

13.2 Filter modul chart



Figur 13.2: Her er et flowchart over den del af vores løsningsforslag som vi har implementeret i SML

13.3 Kildekode

Kildekoden til smlnaivebayes kan findes på:

`http://www.but.auc.dk/~arme0233/SMLBAYES/smlnaivebayes.tar.gz`

Kapitel 14

Interview teknik

De følgende afsnit i dette kapitel er skrevet med bistand fra[12]

14.1 Generel teknik

Interviewteknik er synonym med kunsten at lytte og spørge. Det vigtigste, man kan gøre, for at blive accepteret af den man skal interviewe, er at udstråle interesse og nysgerrighed, det vil med andre ord sige, at man skal have sat sig grundigt og helst oprigtigt ind i det stof, man skal interviewe om. Spørgsmål kan opdeles i typer, hvor den vigtigste sondering går mellem åbne og lukkede spørgsmål. Man skal gøre sig klart hvilke omgivelser man skal interviewe i, og dermed forsøge at undgå at vække for meget opmærksomhed om sin egen person.

14.1.1 Før interviewet

Først er det nødvendigt at sætte sig godt ind i det stof, man vil interviewe om, lave en liste over de emner eller temaer, man vil kunne tale med svareren om, og udvælger dem der er bedst egnede til at få ens budskab frem. Man kan evt. nedskrive spørgsmålene til de enkelte emner eller temaer. Herefter skal man gøre op med sig selv, hvilken registreringsteknik der skal bruges. Under interviewet er det krævende både at tage notater, lytte intenst, tænke sig om og spørge, det er mere bekvemmeligt at optage det. Der er dog både fordele og ulemper ved begge muligheder. Man skal sætte sig mentalt op til interviewet. Herunder skal man tage højde for at forskellige uventede situationer kan opstå, hvor det er vigtigt at forholde sig afklaret med den sitiuation man er i. Eksempler kunne være: svareren sidder og lyver eller giver udtryk for abnormale holdninger.

14.1.2 Under interviewet

Man skal forsøge at styre interviewet henimod de emner man ønsker belyst, ved at komme med de rette spørgsmål. Det er ikke anbefalelsesværdigt at have sine spørgsmål nedskrevet, da man let kan miste dialektiken, fordi man koncentrer sig for meget om det næste spørgsmål, istedet for at lade de improvisatoriske impulser styre. Under interviewet veksles der mellem åbne og lukkede spørgsmål, som en god afslutning på et interview kan man udnytte de lukkede spørgsmåls stopklodsvirkning, til at skifte til, at sige tak for interviewet, f.eks. det lyder alt i alt som om universitetet gør sit bedste for at regulere spam mails.

14.1.3 Efter interviewet

Redigeringen af et interview er uden facit liste, det er op til intervieweren at vurdere hvad der er mest relevant og spændende at fremdrage. Man skal tænke på at man griber ind i et andet menneskes liv ved at interviewe om det, derfor må man overveje, hvilke konsekvenser denne indgriben kan få, det er derfor af største vigtighed, at man giver den interviewede mulighed, for at komme med indsigelser mod det man ønsker at offentligøre.

14.2 Interview med en system administrator

Torsdag den 211102, var vores gruppe indbudt til at medvirke til et interview på Aalborg universitet. Vores opponent gruppe,som også har "Spam skal bekæmpes"som projekt emne, havde arrangeret et interview med Wladyslaw Pietrazek på Frederiksbajersvej. Wladyslaw er Universitetes system administrator og har suverænt kontrollen over at håndhæve og iværksætte Universitetets ønskede restriktioner mht.elektronisk post.
Formålet med dette interview, var at få belyst, i hvor stort et omfang universitetet bliver bombarderet med spam, og hvilke forebyggende foranstaltninger, der bliver taget for at formindske dette. Vores opponent gruppe havde udarbejdet en liste med spørgsmål, som de fandt relevante at stille. Interviewet, der tog en god time, optog vi via en mikrofon, der var tilsluttet en bærbar computer. Vi afspillede optagelsen efter hjemkomsten, vi opdagede at lyd kvaliteten ikke var den bedste, men var alligevel i stand til at opfange ca 95 procent.
Proceduren for en indkommende e-mail til Aalborg Universitet er følgendet: Det bliver, via RBL's blackliste, checket om mailen kommer fra en åben server, gør den det, bliver den automatisk afvist, der bliver ikke kigget efter hvem der skulle modtage den, eller hvem der har sendt den. Slipper e-mailen igennem RBL's blackliste sortering, bliver den kørt igennem spamfiltret Spamassasin, der har et tilhørende blackliste over kendte spammere. Risikoen for fejlafvisninger er stor, men det er et spørgsmål om prioritet, for der bliver sendt meget spam med de åbne servere. De fleste ISPer siger at deres klienter skal sende deres mail, igennem deres mail servere, ISPerne tjekker om afsender adressen er gyldig, altså om der kan svares til den, kan der ikke det, bliver mailen slettet, men som Wladyslaw siger, der bliver opfanget og slettet en masse spam, men der kommer fortsat flere

og flere. Spammerne er kreative og er begyndt at stjæle gyldige e-mail adresser, til at sende deres spam med. Aalborg Universitets e-mail adresse bliver brugt til at spamme med, det er meget svært at stoppe den slags. Det er strafbart at skjule sin afsender identitet, på Aalborg Universitet er det minimum 30 dages bortvisning, og i svære tilfælde er det en permanent bortvisning. Det har kostet Universitetet ca. 10 millioner at lave deres central og det koster ca 2 millioner pr.år at vedligeholde inkl.deres Internetforbindelse over Sverige.

14.3 Konklusion af Interview

Vores gruppes debut i interviewer rollen, blev betrådt med de karakteristiske kendetegn for en god interviewer, nemlig med en stor interesse for emnet og med en nysgerrighed, der lyste ud af vores øjne. Wladyslaw viste sig at være en yderst kompetent og meddelsom herre inden for hans fagområde. Han var tilmed temmelig ukompliceret at være sammen med, da man kunne mærke at han brændte for at fortælle os det, vi ønskede at vide. Vi var ikke optimalt forberedt på interviewet, da vi ikke havde sat os ind i interview teknik, men forløbet gik ud over alt forventning. Vores udbytte af inteviewet betegner vi som optimalt, da vi fik svar på det vi havde forberedt og mere til. Efter vores insigt i interview teknik, burde der være ting, såsom spørgsmål eller handlinger vi gerne ville gøre om, for at optimere udbyttet af interviewet, men det er ikke tilfældet. Vi tillægger dette faktum den betydning, at Wladyslaw meget nær var den fuldendte svarer. Vores indsigt i interview teknik vil, ved et evt. nyt interview, komme os tilgode, da vi ikke kan forventet, at det går lige så problemfrit hver gang.

14.4 Eksempel på naiv bayesiansk sandsynlighed-regning

```
               Token          spam      nospam
              ----------------------------------
50 mails       fisk            11          5
25 er spam     ost              2         20
               lam             22         20
```

Mailen vi skal undersøge indeholder ordene : "fisk"og "ost"
dvs.

$$P(C = SPAM|fisk = 1, ost = 1, lam = 0) = \frac{P(fisk = 1, ost = 1, lam = 0|C = SPAM)}{P(fisk = 1, ost = 1, lam = 0)} \tag{14.1}$$

$*P(C = SPAM)$

1. Første del:

$$P(fisk = 1|C = SPAM) = \frac{count(fisk = 1, C = SPAM)}{count(C = spam)} = 11/25$$
$$P(ost = 1|C = SPAM) = \frac{count(ost = 1, C = SPAM)}{count(C = spam)} = 2/25$$
$$P(lam = 0|C = SPAM) = \frac{count(lam = 0, C = SPAM)}{count(C = spam)} = 3/25$$

Hvilket giver:

$$P(fisk = 1, ost = 1, lam = 0|C = SPAM) \qquad (14.2)$$
$$= (11/25) * (2/25) * (3/25)$$
$$= 0.0042$$

2. Anden del:

$$P(C = SPAM) = \frac{count(C = SPAM)}{count(C)} = 25/50 = 0.50 \qquad (14.3)$$

3. Tredie del:

$$P(fisk = 1) = \frac{count(fisk = 1)}{count(C)} = 16/50$$
$$P(ost = 1) = \frac{count(ost = 1)}{count(C)} = 22/50$$
$$P(lam = 0) = \frac{count(lam = 0)}{count(C)} = 8/50$$

Hvilket giver:

$$P(fisk = 1, ost = 1, lam = 0) = \qquad (14.4)$$
$$(16/50) * (22/50) * (8/50) = 0.0225$$

4. Samlet:

$$P(C = SPAM|fisk = 1, ost = 1, lam = 0) = \qquad (14.5)$$
$$(0.0042/0.0225) * 0.50 = 0.0937$$