Prosjektoppgave i AST2210 – Analyse av fire-års COBE-DMR data

Hans Kristian Eriksen og Tone Melvær Ruud

1 Introduksjon

Vi skal i denne obligatoriske oppgaven gjenta analysen av fire-års COBE-DMR-dataene som ble publisert i 1994, og vi vil fokusere på de to viktige kosmologiske parameterne Q og n, tilsvarende *amplituden* og *hellningen* til CMB-spekteret. Analysen skal beskrives i form av en Astrophysical Journal Letters (ApJL) artikkel, og det er denne artikkelen som vil være det endelig produktet i oppgaven, og som skal leveres inn.

Før man begynner arbeidet, kan det være lurt å lese artikkelen av Górski et al. (1994), som er tilgjengelig i katalogen "∼hke/AST2210/cobe_project" – dere skal i grunnen plagiere denne artikkelen :-) Alle data- og kode-filer er også tilgjengelige i samme katalog.

Det kan også være lurt å lese artikkelen av Eriksen et al. (2007), tilgjengelig i samme katalog, fordi denne har en struktur som ligger nærmere opp til den dere skal bruke. Mens Górski et al. (1994) har en følge-artikkel som beskriver metoden, og den vedlagte kun beskriver resultatene, så skal dere både beskrive metode, data og resultater i samme artikkel. Eriksen et al. (2007) er et eksempel på denne strukturen, med et tema som er nært beslektet med den aktuelle prosjekt-oppgaven.

1.1 Bakgrunn: Kosmisk bakgrunnsstråling (CMB)

I denne oppgaven bruker vi observasjoner av fenomenet kjent som kosmisk bakgrunnsstråling, eller *Cosmic Microwave Background radiation* (CMB) på engelsk. Før vi går i gang med analysen av disse dataene, er det lurt å forstå hva dette er, derfor denne raske oppsummeringen:

Kort fortalt består CMB av de eldste fotonene i Universet. De oppsto ca. 300 000 år etter Big Bang, da det unge Universet hadde ekspandert og kjølnet nok til at elektroner og protoner kunne bindes sammen til nøytralt hydrogen, uten umiddelbart å rives fra hverandre igjen (slik det var opp til dette punktet, på grunn av den høye tettheten og temperaturen i gassen). I en ionisert gass (altså gass med frie elektroner) kan ikke fotoner bevege seg særlig langt uten å kollidere med et elektron slik at det skifter retning. Vi kan derfor si at det ioniserte, unge Universet var opakt (ikke gjennomsiktig). Men ved rekombinasjon, som er det vi kaller hendelsen da nøytrale atomer oppsto for første gang, ble Universet

brått gjennomsiktig – fotonene i gassen kan strømme fritt, uten å kollidere med noe som helst.

På grunn av Universets ekspansjon blir disse fotonene rødforskjøvet, dvs. at bølgelengden deres blir lengre ettersom tiden går. Da de ble dannet hadde de bølgelengder tilsvarende temperaturen i gassen på det tidspunktet, altså ca 3000 K. Rødforskyvningen gjør at de CMB-fotonene vi kan observere i dag har en temperatur på ca. 2.7 K, noe som tilsvarer mikrobølger (derav navnet). Da de ble dannet, var fotonene jevnt fordelt i hele det unge Universet: De var overalt og bevegde seg i alle mulige retninger. Slik er det fortsatt, og vi kan derfor observere bakgrunnsstrålingen i form av fotoner med (tilnærmet) nøyaktig samme energi som kommer mot oss fra alle kanter.

Siden Universet har så lav tetthet, har de fleste av fotonene som "slapp fri" ved rekombinasjonen, strømmet (tilnærmet) uforstyrret gjennom Universet helt frem til i dag. Det betyr at vi, ved å detektere denne strålingen, kan få informasjon om forholdene da de ble dannet – et "babybilde" av Universet. CMB er derfor en av de aller viktigste observable størrelsene i kosmologien. De siste drøyt 20 årene, etter gjennombruddet med COBE, har forskningen i stor grad vært fokusert på én viktig egenskap ved CMB, nemlig at den er *anisotrop*, altså *ikke* nøyaktig lik i alle retninger. Ved å måle de ørsmå forskjellene i temperatur i ulike retninger, kan vi lære om hvordan gassen i Universet var fordelt ved rekombinasjon: Noen steder var det litt varmere (høyere tetthet), og andre steder litt kaldere (lavere tetthet), og fotonene fikk derfor tilsvarende forskjeller i temperatur. De små forskjellene i tetthet utviklet seg med tiden, under påvirkning av gravitasjon, til strukturene vi ser i dag: Galakser og galaksehoper, og store tomrom. Informasjonen om hvordan Universet så ut på et tidlig tidspunkt kan vi bruke til å skille mellom ulike modeller for hvordan det har utviklet seg, og disse observasjonene har derfor vært (og fortsetter å være) et svært viktig bidrag til vår forståelse av verden.

1.1.1 Om sfærisk-harmonisk dekomposisjon og powerspektra

Til slutt, litt om de matematiske verktøyene vi skal bruke videre: Som nevnt over er vi interessert i temperaturen til bakgrunnsstrålingen i forskjellige retninger på himmelen, nærmere bestemt hvor mye temperaturen i hvert punkt avviker fra gjennomsnittet. Vi avbilder disse avvikene i form av himmelkart (vanligvis i Mollweide-projeksjon, altså på en oval flate, som et vanlig verdenskart) som viser blå og røde flekker i størrelsesorden $\pm 100\,\mu$K. Siden disse avvikene har sitt opphav i fenomener vi trenger statistikk for å beskrive (fordeling av partikler i gass) er det først og fremst de statistiske egenskapene til kartet vi er interessert i. Et viktig verktøy her er sfærisk-harmonisk dekompisisjon.

De sfærisk-harmoniske funksjonene $Y_{\ell m}$ er sfæriske bølgefunksjoner. De kan brukes som basisfunksjoner for å dekomponere et hvilket som helst felt som er definert på en kuleflate, og brukes hyppig i mange områder av fysikk. Dere har antakelig allerede brukt dem i FYS2140. Dekomponeringen kan ses som en todimensjonal analog til Fourier-transformasjon: Den "sorterer" signalet vårt i

en serie bølger med varierende bølgelengde,

$$\Delta T(\hat{n}) = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(\hat{n}), \tag{1}$$

der $a_{\ell m}$-ene er amplituder som forteller oss hvor mye av hver bølgefunksjon vi trenger for å bygge opp det totale signalet. Det viser seg at den interessante informasjonen lar seg oppsummere i form av et *powerspektrum*, C_ℓ, som gir variansen til $a_{\ell m}$-ene som funksjon av ℓ: [1]

$$\langle a_{\ell m} a^*_{\ell' m'} \rangle = \delta_{\ell \ell'} \delta_{m m'} C_\ell \tag{2}$$

Det er ved hjelp av powerspektret vi knytter teori og observasjon sammen: Ulike modeller for Universets utvikling gir litt forskjellige prediksjoner for funksjonsformen til C_ℓ, og ved å sammenlikne powerspektra fra CMB-kartene våre med de teoretiske kurvene finner vi ut hvilke modeller som stemmer best overens med virkeligheten.

2 Maximum-likelihood-analyse av COBE-DMR

2.1 Beskrivelse av data

COBE (*COsmic Background Explorer*) var en satelitt som ble skutt opp av NASA i 1989 for å måle egenskapene til den kosmiske bakgrunnsstrålingen. De mest slående resultatene fra eksperimentet var målingene som viste at frekvensspekteret til strålingen var et nær perfekt sortlegemespektrum (gjort av instrumentet FIRAS), og de første målingene av fluktuasjoner (anisotropi) (gjort av instrumentet DMR). Til sammen resulterte disse målingene i Nobel-prisen i fysikk i 2006 for lederene av prosjektet, John Mather og George Smoot.

Vi skal i denne oppgaven gjenta analysen av DMR-dataene. Spesielt skal vi benytte observasjonene tatt på 53 og 90 GHz, som er de "reneste" (dvs., minst støy og galaktiske forgrunner) av DMRs tre kanaler. Det ene av disse kartene er vist i figur 1.

Videre trenger vi å vite usikkerheten til dataene. Vi antar (og du kan trygt anta at noen har sørget for å sjekke at denne antakelsen er god nok) at hver observerte piksel, på grunn av instrumentell støy, kan tenkes på som en sample fra en Gaussfordeling sentrert på den "sanne" pikselverdien, og med et visst standardavvik. Standardavviket per piksel er oppgitt i et eget *RMS-kart* (root-mean-square). RMS-kartet for 90 GHz-kanalen er også vist i figur 1. Standardavviket forteller oss hvor sikker vi er på observasjonen vår: Jo mindre verdi, jo smalere er Gausskurven for en gitt piksel. Og jo mer observasjonstid et punkt på himmelen har fått, jo lavere blir RMS-en, altså standardavviket, for denne pikselen. RMS-kartet kan derfor også fortelle oss hvilke områder som har fått mest og minst observasjonstid.

[1] De $2\ell+1$ $a_{\ell m}$-ene for en gitt ℓ har alle samme varians, så vi trenger ikke noe m-avhengighet i powerspektret. Fysikken bak dette lærer dere mer om i senere kurs, AST3220 Kosmologi 1 og/eller AST5220 Kosmologi 2.

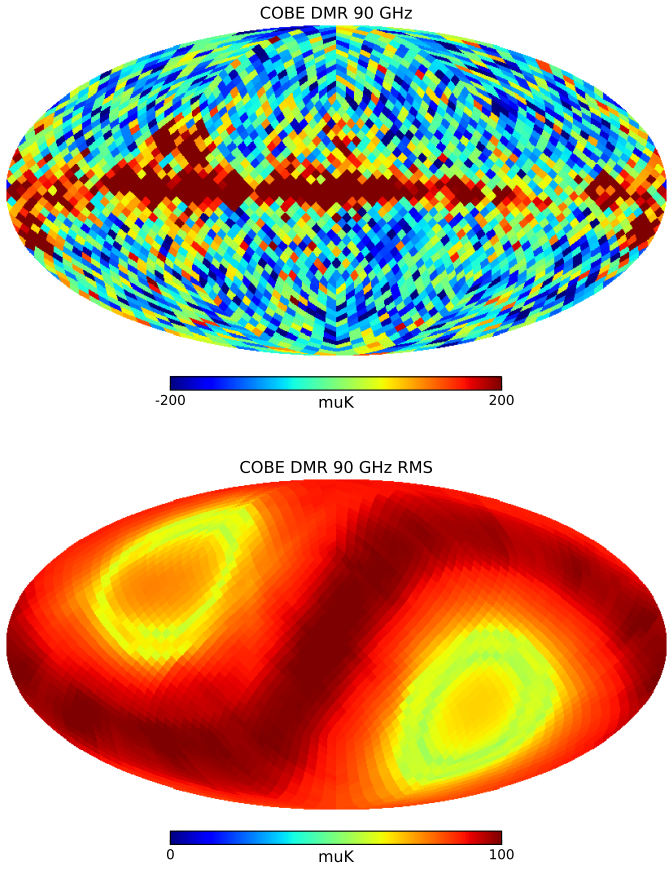

Figure 1: Kart og tilhørende RMS-kart av COBE-DMR 90 GHz, det ene av de to datasettene vi skal analysere. Merk at disse kartene viser hele himmelen, projisert på en flate (Mollweide-projeksjon). De er i galaktiske koordinater, sentrert på retningen mot Melkeveiens sentrum. Det røde "båndet" på tvers av kartet er galaktisk forgrunnsstråling, som naturlig nok er kraftigst langs galakseplanet. RMS-kartet (nederste panel) viser støyegenskapene til hver av pikslene i det øverste panelet. Grunnen til at det ikke er likt over hele himmelen er at noen områder har fått mer observasjonstid enn andre. Den mørkerøde sonen tilsvarer ekliptikken (planet til vårt solsystem): Dette området har man viet minst observasjonstid, på grunn av de mange kraftige strålingskildene som befinner seg her.

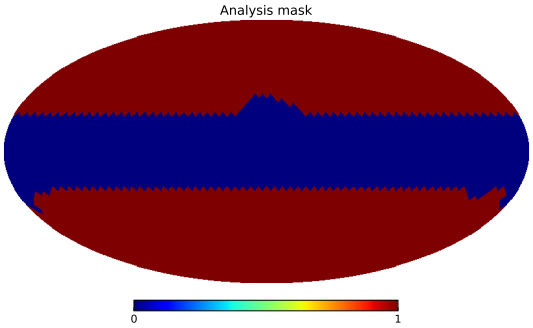

Figure 2: Analysemasken. De røde pikslene er de vi tar med i analysen. De blå er områder nær galakseplanet, som kan ha forgrunnssignal, og som vi derfor fjerner fra analysen for å være på den sikre siden.

Som vi ser i 90 GHz-kartet, er det noen områder av himmelen som er dominert av *forgrunner*, altså sterke strålingskilder som ligger i synsfeltet vårt og "overdøver" CMB-fotonene. Forgrunnene på disse frekvensene er først og fremst stråling fra støv i galaksen vår, og forgrunnene er derfor sterkest i områdene nær galakseplanet. For å være sikre på at vi ikke får feil resultat av analysen på grunn av slike forgrunner, bruker vi en *maske* som definerer hvilke piksler vi ikke stoler på. Figur 2 viser masken vi skal bruke nå.

Det neste vi må ta høyde for er den instrumentelle *beamen*. Dette er en funksjon som forteller hvor stort område på himmelen instrumentet vårt ser til enhver tid. Ikke noe CMB-instrument observerer kun ett punkt på himmelen av gangen, men derimot en endelig romvinkel. Eventuelle variasjoner i den sanne himmeltemperaturen innenfor denne romvinkelen kan vi ikke se, fordi alt signalet fra dette området puttes i samme piksel. Dette fører i praksis til en "utsmøring" av signalet, og de minste skalaene blir filtrert bort. Matematisk beskriver vi denne operasjonen som en konvolvering i piksel-rommet, eller en multiplikasjon i harmonisk rom, slik at likning 1 blir erstattet med

$$\tilde{T}(\hat{n}) = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} b_\ell a_{\ell m} Y_{\ell m}(\hat{n}), \qquad \text{beam-konvolvert kart.} \tag{3}$$

Beamen til DMR er kjent, og vi skal ta hensyn til den i analysen vår. Hvordan dette gjøres i praksis vil bli beskrevet senere. Du kan med fordel plotte den, for å se hvordan denne funksjonen ser ut, så du lettere forstår hvilken effekt den har.

I alt består vårt DMR-data-sett av de følgende komponentene:

- To filer med CMB-data observert på 53 og 90 GHz. Totalt 1941 datapunkter med tilhørende retningsvektor. (1131 piksler av totalt 3072 er fjernet ved maskering.) Oppløsningen er på ca. 220' ($\sim 0.1^\circ$)

- To tilhørende filer med standardavviket til hvert datapunkt: Måleusikkerheten i hver piksel er gitt ved en sentrert Gaussisk distribusjon med varians σ_p^2, der σ_p er gitt i filene.
- Den instrumentelle beamen, b_ℓ.

Disse er tilgjengelig i katalogen "".

2.2 Modellering av dataene

2.2.1 Generelt om Gaussiske felt

Kanskje den vanligste sannsynlighetsfordelingen man møter i praktisk dataanalyse er normalfordelingen, også kalt Gaussdistribusjonen. I én dimensjon ser den ut som følger,

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \tag{4}$$

der μ er gjennomsnittet til distribusjonen, og σ^2 er variansen. σ alene kalles standardavviket, og måler usikkerheter i målingen; en gitt observasjon ligger innenfor en avstand på 1σ fra det sanne svaret med 68% sannsynlighet.

Ofte studerer man sentrerte distribusjoner, dvs. fordelinger med $\mu = 0$, og det skal vi gjøre i det følgende også. Vi ser derfor bort fra denne parameteren fra nå av.

Dersom datasettet man studerer har flere avhengige, Gaussfordelte variable, må man benytte den multivariate normalfordelingen,

$$p(\mathbf{x}) \propto \frac{1}{\sqrt{|\mathbf{C}|}} e^{-\frac{1}{2}\mathbf{x}^t\mathbf{C}^{-1}\mathbf{x}}, \tag{5}$$

der $\mathbf{x}$ er en vektor bestående av alle datapunktene, og $\mathbf{C} \equiv \langle \mathbf{x}\mathbf{x}^t \rangle$, altså forventningsverdien til ytreproduktet av datavektoren med seg selv, kalles kovariansmatrisen[2]. Kovariansmatrisen har samme funksjon som variansen i det éndimensjonale tilfellet, men måler både variansen til hvert datapunkt x_i for seg (oppført i de diagonale elementene C_{ii}), men også hvor sterkt to forskjellige punkter x_i og x_j er korrelert (oppført i C_{ij}).

Hovedpoenget er imidlertid at dersom man kjenner kovariansmatrisen til en Gaussisk distribusjon, så vet man *alt* om distribusjonen. Det er derfor først og fremst denne man må etablere dersom man ønsker å gjøre dataanalyse med Gaussiske variabler.

2.3 Spesialisering til CMB-data

CMB-data er til en svært, svært god tilnærming Gaussisk distribuerte, og vi ønsker derfor å finne et uttrykk for kovariansmatrisen til våre observerte data. Vi må da først modellere dataene våre.

[2] Vi ser bort fra normaliseringskonstanter av typen 2π i dette uttrykket.

Vi beskriver dataene som en sum av en CMB-komponent og en instrumentell støy-komponent,

$$d(\hat{n}) = s(\hat{n}) + n(\hat{n}) + f(\hat{n}). \tag{6}$$

Her er d det observerte signalet i retning $\hat{n}$, s er CMB-signalet, n er støyen, og f er mulige ikke-kosmologiske forgrunnssignaler. Vi antar at ingen av disse tre komponentene er internt korrelerte med hverandre, slik at gjennomsnittene $\langle sn\rangle = \langle sf\rangle = \langle nf\rangle = 0$. (Dvs., bakgrunnsstrålingen påvirker ikke egenskapene til vår egen galakse, eller støyen i instrumentet vårt.)

Kovariansmatrisen til d er derfor gitt ved

$$\mathbf{C} \equiv \left\langle \mathbf{d}\mathbf{d}^t \right\rangle = \left\langle (\mathbf{s}+\mathbf{n}+\mathbf{f})(\mathbf{s}+\mathbf{n}+\mathbf{f})^t \right\rangle = \left\langle \mathbf{s}\mathbf{s}^t \right\rangle + \left\langle \mathbf{n}\mathbf{n}^t \right\rangle + \left\langle \mathbf{f}\mathbf{f}^t \right\rangle \equiv \mathbf{S}+\mathbf{N}+\mathbf{F}, \tag{7}$$

der $\mathbf{S}$ er kovariansmatrisen til CMB-signalet alene, $\mathbf{N}$ er kovariansmatrisen til støyen, og $\mathbf{F}$ er kovariansmatrisen til forgrunnene. Hver av disse er en $N_{\rm pix}\times N_{\rm pix}$ matrise i vårt tilfelle.

Vi kjenner verken til det nøyaktige CMB-signalet eller støykomponenten, men kun deres statistiske egenskaper. For å starte med det enkleste tilfellet, så antar vi at først støyen er Gaussisk og ukorrelert fra piksel til piksel, med standardavvik σ_p. Kovariansmatrisen er derfor

$$N_{ij} = \langle n_i n_j \rangle = \sigma_i^2 \delta_{ij}, \tag{8}$$

der i og j er to piksel-indekser. Med andre ord, støy-kovariansmatrisen er diagonal, med vanlig varians langs diagonalen; den multivariate støydistribusjonen er ganske enkelt produktet av vanlige én-dimensjonale Gaussfordelinger for hver piksel. Dette er en svært god tilnærming for DMR.

Vi antar videre[3] at CMB-feltet er Gaussisk og isotropt, men korrelert fra piksel til piksel. I dette tilfellet er det mulig å vise at piksel-piksel-kovariansmatrisen er

$$S_{ij} = \frac{1}{4\pi}\sum_{\ell=0}(2\ell+1)(b_\ell p_\ell)^2 C_\ell P_\ell(\cos\theta_{ij}), \tag{9}$$

der b_ℓ er den instrumentelle beamen beskrevet over, p_ℓ kalles *piksel-vindu*, og gir effekten av endelig pikselisering (som oppfører seg prinsipielt på samme måte som beamen: Himmelen har småskalavariasjoner vi ikke kan detektere på grunn av den endelige størrelsen til pikslene våre. Plott gjerne denne funksjonen også!), $P_\ell(x)$ er Legendre-polynomene, f.eks. kjent fra matematisk fysikk, og θ_{ij} er vinkelen mellom pikselene i og j.

Legg merke til at det er her kosmologien kommer inn: Powerspekteret C_ℓ er en teoretisk kurve som bestemmes av kosmologiske parametere, og ved å endre disse parametrene, får man forskjellige korrelasjonsegenskaper i CMB-feltet. I denne oppgaven skal vi se på en spesiell klasse av modeller, nemlig de parametrisert ved en amplitude Q og en spektralindeks n ($P(k) \propto k^n$; Bond og Efstathiou 1987)

$$C_\ell = \frac{4\pi}{5}Q^2 \frac{\Gamma(\ell+\frac{n-1}{2})\Gamma(\frac{9-n}{2})}{\Gamma(\ell+\frac{5-n}{2})\Gamma(\frac{3+n}{2})}. \tag{10}$$

[3]Dette er en antakelse med solid fysisk forankring, som dere lærer mer om i AST3220/5220.

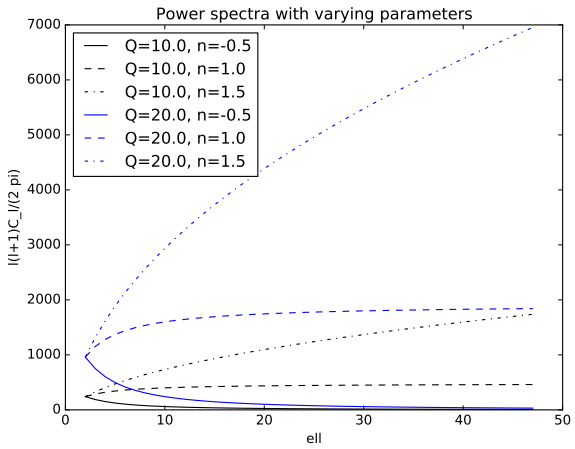

Figure 3: Eksempler på hvilken effekt endring av parameterne Q og n har på powerspektret C_ℓ.

(Valg av den litt spesielle Q-normaliseringen er gjort for å kunne sammenlikne resultatene med de som ble presentert i 1994.) Bruken av Γ-funksjonen får dette til å se veldig komplisert ut. Men denne kan slås opp i matematisk formelsamling eller finnes på nett, og det viktigste her er at den har egenskapen $\Gamma(1+x) = x\Gamma(x)$. Merk også at uttrykket for $\ell = 2$ forenkles til $C_2 = 4\pi/5\, Q^2$. Man kan derfor beregne $C_{\ell+1}$ rekursivt, gitt C_2 som funksjon av Q. Merk videre at man må sette $C_0 = C_1 = 0$ manuelt, fordi vi fjerner monopolen og dipolen (altså bidrag fra dekomponeringen i likning 1 som har $\ell = 0$ eller $\ell = 1$) som beskrevet under. I figur 3 viser vi hvordan powerspektrene i denne klassen ser ut for forskjellige parameterverdier.

Med denne definisjonen har vi koblet CMB-kovariansmatrisen – og dermed våre observerte data – til to enkle inflasjons-parametere Q og n.

Endelig, vi må inkludere et ledd som tar fjerner effekten av mulige monopoler og dipoler i kartene[4]. I praksis gjøres dette ved å legge til ekstra ledd i kovariansmatrisen: Anta at man ønsker å være fullstendig insensitiv til et signal $f(\hat{n})$, altså et eller annet kart – dersom det finnes en slik komponent i kartet, ønsker man at dette skal ha null statistisk vekt. Dette oppnås ved å si at usikkerheten til denne komponenten er "uendelig" stor. I praksis legger man til det følgende leddet til kovariansmatrisen,

$$\mathbf{F} = \lambda \mathbf{f}\mathbf{f}^t. \tag{11}$$

[4]DMR er et såkalt differensielt instrument, hvilket betyr at det kun måler forskjeller mellom temperaturer i to forskjellige retninger, og ikke absolutte verdier. Det betyr at man ikke kan finne gjennomsnittstemperaturen (= monopolen, signalkomponenten med $\ell = 0$) til CMB-signalet ved å studere DMR-data. Derimot må man *marginalisere* over denne komponenten for ikke å forstyrre for andre signaler. Tilsvarende måler DMR heller ikke dipolen (=signalkomponenter med $\ell = 1$), fordi denne er sterkt dominert av en Doppler-effekt pga. vår egen bevegelse gjennom universet. Den virkelige CMB-dipolen er ikke observerbar.

Her er λ en stor konstant (f.eks. 10^3), og $\mathbf{f}$ er et fast signal på himmelen (engelsk: template) man ønsker å være ufølsom for. $\mathbf{f}\mathbf{f}^t$ er ytre-produktet til denne templaten tatt med seg selv. F.eks., dersom man ønsker å marginalisere[5] over monopolen, legger man ganske enkelt til en stor konstant til den totale kovariansmatrisen.

For å oppsummere, den endelige kovariansmatrisen til dataene er gitt ved

$$\mathbf{C}(Q, n) = \mathbf{S}(Q, n) + \mathbf{N} + \mathbf{F}, \tag{12}$$

der de tre individuelle matrisene er definert som over.

2.4 Maximum-likelihood analyse

Det vi ønsker å oppnå med denne analysen, er å finne ut hvilke verdier av modellparameterne, Q og n, som gir den beste overensstemmelsen med datasettet vårt. Det finnes i prinsippet ikke ett fasitsvar på dette spørsmålet, så her må man velge seg én av flere mulige metoder. En svært vanlig strategi kalles *maximum likelihood*.

Maximum likelihood-analyse tar utgangspunkt i likelihood-funksjonen, som er identisk med den samlede sannsynlighetsfordelingen for observerte data, gitt en modell, men tolket som en funksjon av modellparametrene:

$$\mathcal{L}(Q, n) = p(\mathbf{d}|Q, n)$$

Denne funksjonen forteller hvor høy sannsynlighet vi har for å observere det vi faktisk har observert, under ulike modeller. Filosofien bak maximum likelihood-analyse er at vi, ved å finne toppunktet til denne funksjonen, finner de parameterne som beskriver datasettet vårt best. (Som ikke er ensbetydende med at de er de "sanne" parameterverdiene – hvis du vil ha en utredning om temaet, gå og finn en statistikkprofessor.)

Den samlede fordelingen vi trenger her, er bare den multivariate Gauss-fordelingen gitt i likning 5, med datavektoren $\mathbf{d}$ som variabel. Når man så har konstruert kovariansmatrisen, så er resten av analysen ganske rett fram. Basert på denne fordelingen følger det at *log-likelihooden* lyder

$$-2 \log \mathcal{L}(Q, n) = \mathbf{d}^T \mathbf{C}^{-1} \mathbf{d} + \log |\mathbf{C}| + \text{konstant}, \tag{13}$$

og oppgaven er da ganske enkelt å beregne denne for forskjellige verdier av (Q, n).

Merk at det er vanlig å benytte log-likelihooden istedet for selve likelihooden, siden dette er et mer håndterlig tall: Med 1000 piksler vil log-likelihooden være av størrelsesorden 1000, mens selve likelihooden er $\mathcal{O}(e^{-1000})$. Numeriske feil blir da problematiske i naturlige enheter. Det er derfor et godt tips å gjøre alle

[5] Å marginalisere er å fjerne én variabel fra en felles distribusjon, for å gå fra en betinget til en ubetinget fordeling. Hvis vi har en felles distribusjon $f(x, y)$, og ønsker å finne distribusjonen av x alene, kan vi marginalisere ut y, som betyr at vi integrerer over alle mulige verdier av y: $f(x) = \int f(x, y)\, dy$

beregninger i logaritmer, og så først eksponentiere helt til slutt, om nødvendig. Man kan også se bort fra konstant-leddet over, og heller velge å normalisere funksjonen til slutt, om det skulle trengs.

Hovedproduktet i dette prosjektet er en to-dimensjonal figur av likelihooden for (Q, n). I dette tilfellet har vi et lite nok datasett til at brute-force grid-evaluering, dvs. at vi beregner verdien av $\log \mathcal{L}$ på et grid av mulige verdier, er gjennomførbart. For større datasett er man nødt til å bruke mer sofistikerte metoder, som Markov Chain Monte Carlo. Slike metoder er pensum i senere kurs, f.eks. FYS3150 – Computational physics. (Hvis du har tid og lyst står du selvsagt fritt til å prøve å løse oppgaven med MCMC.)

Man skal også beregne de marginale best-fit verdiene for hver av de to parameterene, med tilhørende usikkerheter. Matematisk får man de en-dimensjonale distribusjonene ved å integrere over den andre, f.eks.:

$$\mathcal{L}(Q) = \int \mathcal{L}(Q, n) dn. \tag{14}$$

Ved bruk av MCMC er dette trivielt, siden man da ganske enkelt kan lage en-dimensjonale histogrammer fra MC samplene direkte. Dersom man gjør en grid-beregning, må man faktisk utføre integralet numerisk. I begge tilfeller ønsker man å komme fram til et svar av formen $n = 1.1 \pm 0.2$.

2.5 Implementasjon

Man kan i utgangspunktet velge hvordan oppgaven skal gjøres selv, men det enkleste er helt klart å benytte seg av Python-malen som er gjort tilgjengelig i standard-katalogen. Denne malen er en nær komplett versjon av det endelige programmet, men noen få kritiske kommandoer er fjernet.

Programmet består av to filer, nemlig hovedprogrammet ("cmb_likelihood.py") og en hjelpemodul ("cmb_likelihood_utils.py"). Hovedprogramfilen er helt komplett, og man trenger ikke å gjøre noe som helst med denne, dersom man ikke vil. Men man må definitivt lese gjennom den, og sørge for at man forstår hva som skjer på de forskjellige stadiene. En annen ting er output-statements – dersom man ønsker et annet format på resultat-filene, må dette gjøres i hovedprogrammet.

Hjelpe-modulen er derimot ikke helt ferdig. Alle subrutiner er på plass, men noen få kommandoer mangler. Hva som mangler er oppgitt i begynnelsen av modulen. Det er også markert i hver enkelt rutine hva som skal fylles inn hvor, men ikke nødvendigvis hvordan.

2.5.1 Programmeringstekniske tips

Det er to operasjoner i dette prosjektet (ihvertfall så lenge man bruker Python) som kan bli svært tidkrevende, og hvor det derfor blir viktig å programmere lurt. Dette gjelder beregningen av signal-kovariansmatrisen $\mathbf{S}$ (likning 9) og den ferdige log-likelihooden (likning 13).

Kovariansmatrisa: Gitt uttrykket for $\mathbf{S}$ er det lettvint å sette opp en trippel for-loop for å beregne denne matrisa. Men dette tilfellet er et kremeksempel på at for-løkker i Python kan være trege greier, og derfor er det alltid lurt å vektorisere Pythonkode når det er mulig. Heldigvis har vi NumPy, som er et bibliotek basert på C, og som er optimalisert for array-operasjoner. Her bør man derfor ta litt ekstra tid og se på uttrykket med "vektor-briller": Det kan nemlig ses som en serie med elementvise vektor-multiplikasjoner, etterfulgt av et indreprodukt (som utgjør summen over ℓ). Du kan med fordel prøve begge metodene, både for å se hvor mye fortere koden går med vektorisering, og for å sjekke at de gir samme svar (fint for debugging).

Log-likelihood: Uttrykket for $\log \mathcal{L}$ innbyr også til en selvsagt løsningsstrategi: Invertér $\mathbf{C}$ med f.eks. scipy.linalg.inv, og beregn determinanten med f.eks. scipy.linalg.det eller numpy.linalg.slogdet. Invertering og determinant-beregning for store matriser er tunge oppgaver, heldigvis er det folk som har taklet disse problemene før oss, og det finnes brukervennlige biblioteker for formålet (LAPACK er en utbredt variant, som SciPy kjører back-end i disse rutinene). Men i praksis er det ikke så ofte man trenger å invertere en matrise, det finnes svært ofte triks som gjør at man kan unngå problemet. I dette tilfellet er nøkkelen Cholesky-dekomposisjon.

Cholesky-dekomposisjon er å uttrykke en matrise som et produkt av en triangulær matrise (i dette tilfellet nedre triangulær, men øvre kan også brukes - da ser uttrykkene litt annerledes ut) og dennes transponerte,

$$\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathrm{T}} \quad \Rightarrow \quad \mathbf{A}^{-1} = (\mathbf{L}^{-1})^{\mathrm{T}}\mathbf{L}^{-1}.$$

Med denne dekomposisjonen får man determinanten "med på kjøpet", siden vi har

$$\det \mathbf{A} = \det \mathbf{L} \, \det(\mathbf{L}^{\mathrm{T}}) = (\det \mathbf{L})^2,$$

hvor determinanten til en triangulær matrise bare er produktet av diagonalelementene.

Å løse et likningssett av typen

$$\mathbf{L}\mathbf{x} = \mathbf{y}$$

for $\mathbf{x}$ er veldig enkelt, og det går langt raskere enn å invertere en matrise, selv med bruk av LAPACK. Trikset her er derfor å omformulere

$$\mathbf{d}^T\mathbf{C}^{-1}\mathbf{d}, \quad \text{der} \quad \mathbf{C}^{-1} = (\mathbf{L}^{-1})^{\mathrm{T}}\mathbf{L}^{-1},$$

til noe på formen

$$\mathbf{x}^T\mathbf{x},$$

der $\mathbf{x}$ er løsningen til et triangulært likningssett av typen vist over.

SciPy-rutinene scipy.linalg.cholesky og scipy.linalg.solve_triangular vil komme godt med her.

Dersom du skulle stå fast, så ikke nøl med å spørre Hans Kristian enten i person eller per email (h.k.k.eriksen@astro.uio.no): Ikke bruk lang tid på å finne ut av et problem, hvis det tar to minutter å spørre en annen! :-)

2.6 Data

De nødvendige dataene ligger i standard-katalogen gitt i begynnelsen av oppgaveteksten:

- cobe_dmr_53GHz_n16.npy(.dat)– DMR-kart og RMS-verdier for 53GHz-kanalen, i form av en liste med umaskerte pikselverdier og tilhørende retningsvektor
- cobe_dmr_90GHz_n16.npy(.dat)– DMR-kart og RMS-verdier for 90GHz-kanalen, i form av en liste med umaskerte pikselverdier og tilhørende retningsvektor
- cobe_dmr_beam.npy(.dat)– beam-funksjonen b_ℓ til DMR, gitt for hver multipol ℓ.
- pixwin_n16.npy(.dat)– piksel-vindu-funksjonen til Healpix ved $N_{\rm side} = 16$.
- params.py – parameter-fil til cmb_likelihood-programmet

Resultater skal genereres og vises for både 53 og 90GHz-kanalene.

Merk at datafilene er gitt i to ekvivalente formater: Binære NumPy-array-filer, og ordinære tekstfiler. For de som velger å løse oppgaven i Python, er det første formatet å anbefale, men .dat-filene er tilgjengelige dersom noen skulle ønske å bruke et annet språk.

2.7 Plotting av resultater

Som alltid, hvordan man velger å lage likelihood-kontur-figuren er opp til en selv. Det er imidlertid laget en svært enkel kode som fungerer som et utgangspunkt. I "work"-katalogen ligger det et script som heter "plot_contours.py", som tar resultat-filen fra cmb_likelihood, og viser det tilsvarende kontur-plottet på skjermen. Hvordan man skriver til fil, og eventuelt gjør figuren penere, må man finne ut av selv, eller spørre med-studenter. Eller man kan benytte helt andre metoder, om man ønsker det.

3 Del 2 – "Publisering" i Astrophysical Journal Letters

Selve programmeringen som skal gjøres i dette prosjektet er ganske enkel, og består av å fylle inn noen få, enkle kommandoer i et nesten ferdig program. Videre vil kjøringen ta noen timer, kanskje opp til et døgn, avhengig av hvor mange grid-punkter man ønsker.

Hoved-arbeidet i prosjektet ligger i å skrive artikkelen. Malen som skal brukes er gjort tilgjengelig i standard-katalogen, og inneholder allerede kapittel-overskrifter, klasse-definisjoner etc. Dersom man ønsker, kan man naturligvis endre disse, men det forventes at man holder seg innenfor den stilen som er vanlig i litteraturen.

Tenk deg at du skriver denne artikkelen for første gang i 1994, og ikke i 2016. Med andre ord, ikke anta at man har resultatene fra WMAP og andre eksperimenter når du beskriver resultatene. Husk at dette er første gang de første fluktuasjonene i universet oppdages. Med andre ord, ikke vær altfor beskjeden – dette kommer til å resultere i en Nobel-pris om 15 år – men bruk likevel et nøkternt språk.

Artikkelen skal inneholde de følgende elementene:

1. Et abstract som svært kort oppsummerer problemet som studeres, metodene og resultatene
2. En introduksjon som gir settingen til analysen – hvorfor gjør vi dette, hva sier teorien om problemet, og er noe gjort før?
3. En metode-seksjon som beskriver data-modellen (med definisjon av kovariansmatriser), likelihood, og algoritme for å kartlegge denne.
4. Beskrivelse av data: Hvilke data er benyttet, hvilke frekvenser, hvor høy oppløsning etc.
5. Resultater:
   (a) Både 53 og 90GHz-kartene skal analyseres
   (b) 2D-kontur-plott skal vises for begge tilfeller, muligens overplottet på hverandre om ønskelig
   (c) Best-fit verdier skal føres i en tabell, sammen med tilhørende usikkerheter
   (d) Detaljer angående analysen (som f.eks. kjøretid etc.) kan også oppføres.
6. Konklusjon: Hva er funnet, og hvilken betydning har disse resultatene? Hvilke eksperimenter bør gjøres i framtiden for å bringe denne forskningen videre?

Endelig, dersom noe er uklart, direkte feil eller noe trenger å forbedres med enten kode, beskrivelse eller maler, så ikke nøl med å si ifra! Alle tilbakemeldinger mottas med stor takk!