

Sparse Bayesian Learning for Interpolation of Radar Volumes

Heikki Myllykoski

School of Science

Thesis submitted for examination for the degree of Master of
Science in Technology.

Helsinki 24.4.2023

Supervisor

Prof. Nuutti Hyvönen

Advisor

PhD Seppo Pulkkinen,
PhD Terhi Mäkinen

Copyright © 2023 Heikki Myllykoski



Author Heikki Myllykoski

Title Sparse Bayesian Learning for Interpolation of Radar Volumes

Degree programme Mathematics and operations research

Major Applied mathematics

Code of major SCI3053

Supervisor Prof. Nuutti Hyvönen

Advisor PhD Seppo Pulkkinen,
PhD Terhi Mäkinen

Date 24.4.2023

Number of pages 50

Language English

Abstract

Weather radars are effective tools for detection of precipitation patterns and weather related hazards. In order to be able to use such instruments effectively, one must adjust for the physics of the measurement, and possibly filter certain echoes out before further analysis. Also, as a radar measures only sections of the atmosphere, gaps between the radar rays are formed. To fill the filtered values and to estimate the values between the measured sections, one uses different methods for interpolation.

This work presents sparse Bayesian learning for interpolation of radar volumes to Cartesian rasters. The benefit of the method is that it provides uncertainty estimates for its fit, and it uses regularization to generalize the interpolation.

The method is compared with simple volume interpolation methods on a selected test volume with three different interpolation cases. The volume data is first preprocessed using a method for attenuation correction and a method to filter out non-meteorological echoes. Method itself is introduced using a type-II discrete cosine transform and conjugate gradient method for inversion as part of an expectation-maximization algorithm. Also, an approximation of the log-likelihood function is presented as a possible way to assess the convergence of the iteration of the expectation-maximization algorithm.

The results imply that the method has problems with convergence due to ill-conditioning of certain involved matrices. This leads to a method that is slow, and hence the number of iterations can not be increased too much. Thus, the method can not be iterated sufficiently for convergence and hence it fails to produce better interpolations than simple methods like inverse distance weighting. The work also presents possible ideas for developing the method based on the obtained results, and observations.

Keywords Bayesian, sparse, learning, radar, weather, classification, attenuation, precipitation, transform, conjugate, gradient, volume, interpolation, atmosphere, atmospheric, science

Tekijä Heikki Myllykoski

Työn nimi Tutkamittausten interpoloiminen bayesilaisittain harventavalla oppimisella

Koulutusohjelma Mathematics and operations research

Pääaine Applied mathematics**Pääaineen koodi** SCI3053

Työn valvoja Prof. Nuutti Hyvönen

Työn ohjaaja FT Seppo Pulkkinen,
FT Terhi Mäkinen

Päivämäärä 24.4.2023**Sivumäärä** 50**Kieli** Englanti

Tiivistelmä

Säätutkat ovat tehokkaita työkaluja sadealueiden ja säähän liittyvien vaarojen havaitsemiseksi. Jotta tutkia käytettäisiin järkevästi, tulee mittausten fysiikka ottaa huomioon ja suodattaa osa saaduista tutkakaiuista ennen kuin mittauksia analysoidaan syvemmin. Koska säätutka mittaa lähtökohtaisesti vain osia ilmakehästä, muodostuu mittausten väliin aukkoja. Suodatettujen arvojen tilalle ja mittaamisesta muodostuviin aukkoihin lasketaan uudet arvot erilaisilla interpolointimenetelmillä.

Tässä työssä kokeillaan bayesilaisittain harventavaa oppimista menetelmän mittaustilavuuksien interpoloimiseksi karteesiin hiloihin. Menetelmän etu on se, että intepolaation lisäksi soviteelle saadaan epävarmuusestimaatit ja että menetelmä soveltaa regularisointia interpolaation yleistämiseksi.

Menetelmää verrataan yksinkertaisempiin mittaustilavuuksien interpolointimenetelmiin eräällä mittaustilavuudella kolmessa eri interpolaatiotapauksessa. Aluksi tilavuusdata esikäsitellään vaimennuskorjauksella ja ei-meteorologisten kaikujen suodatuksella. Menetelmä rakentuu tyyppin II diskreetin kosinimuunnoksen ja konjugaatti-gradientti-matriisikäänös menetelmän päälle osana odotusarvo-maksimointi-algoritmia. Lisäksi työssä esitetään keino logaritmisesta uskottavuusfunktion approksimoimiseksi odotusarvo-maksimointi-algoritmin iteraation edistymisen arviointia varten.

Tutkimustulokset osoittavat, ettei menetelmä suppene ongelmitta johtuen matriiseista, jotka ovat häiriöalttiit. Tämä puolestaan johtaa hitaaseen menetelmään, jolloin iteraatioiden määrää ei voi kasvattaa paljoa. Iteraatioiden raskauden takia menetelmää ei voida iteroida suppenemiseen saakka, ja siksi se ei tuota parempia tuloksia kuin yksinkertaisemmat menetelmät, kuten käänteisen etäisyyden menetelmä. Työ esittää myös ideoita menetelmän jatkokehitykselle.

Avainsanat säätutka, bayesilainen, tilastotiede, interpolointi, harventava, oppiminen, sää, sade, muunnos, konjugaatti, tutkamittaus, gradientti, volyyymi, tilavuus, vaimeneminen, tutka, ilma, ilmatiede

Esipuhe

Kiitokset työn ohjaajille Sepolle ja Terhille sekä työn valvojalle Nuutille. Kiitokset myös työskentelyni ja sen rahoituksen mahdollistaneille esihenkilöille ja työkavereille. Ryhmässämme Lähisään ja älyliikenteen tutkimus ja yleisesti Ilmatieteen laitoksella on ollut mukava tehdä diplomityötä, joten suuri kiitos kaikille sen ilmapiirin mahdollistaneille. Kiitokset kavereille ja kotopuoleen työn ulkopuolisen virkistystoiminnan mahdollistamisesta.

Toivottavasti työtä tehdessä opitut asiat ovat eduksi tässä mahdollisuuksien, mysteerien ja mullistusten maailmassa.

Kumpula, 24.4.2023

Heikki Juhani Myllykoski

Contents

Abstract	III
Abstract (in Finnish)	IV
Esipuhe	V
Contents	VI
1 Introduction	1
2 Theoretical background and earlier research	4
2.1 Weather radars	4
2.1.1 Basic measurements	4
2.1.2 Dual-polarization measurements	6
2.1.3 Attenuation	7
2.1.4 Refraction	8
2.2 Interpolation of weather radar data	8
2.2.1 Methods in earlier research	10
2.2.2 Metrics in earlier research	11
2.3 Basics of Bayesian data analysis	11
3 Research material and methods	14
3.1 Dataset and preprocessing	14
3.1.1 Attenuation correction	15
3.1.2 Echo classification	17
3.2 Sparse Bayesian learning	18
3.2.1 Model description	19
3.2.2 Discrete cosine transform basis	23
3.2.3 Matrix inversion and diagonal of the inverse	25
3.2.4 Computation of the cost function	27
3.3 Experimental setup	28
4 Results	31
4.1 Interpolation onto a three-dimensional raster	31
4.2 Interpolation onto a CAPPI raster	35
4.3 Interpolation onto an RHI raster	38
5 Discussion	41
5.1 Ideas for improvements	43
6 Summary	46

1 Introduction

Understanding weather and climate are of crucial importance for society. According to European Environment Agency, over the period from 1980 to 2020 weather and climate related losses resulted in 487 billion Euro in the current 27 EU nations. For example, traffic, agriculture, safety officials, energy management, and individual people benefit from the weather information in daily decision making, and with that information they can prepare for weather related hazards such as heavy rain and flash floods [1, 2].

Forecasting precipitation and hazards related to precipitation events is performed using measurements from the atmosphere, and one of the measurement instruments is the weather radar. Weather radars are useful as they measure large sections of the atmosphere and the observations are available for further analysis in a matter of seconds. Hence, weather radars can be used, for example, to detect storms when they actually are present. Large numerical weather predictions (NWP) are heavy to compute and their initial states are estimated using extensive data assimilation from weather radar measurements, soundings, satellite measurements, and surface observations. With fine-grained raster sizes for short-time forecasts, that is, about 0-6 h into the future in the order of 1 km and 5 minutes in time and space resolutions, extrapolation of radar echoes outperform NWP. To model for example flash floods on a city, fine-grained time and space resolutions are needed for decision making. Weather radar measurements are hence of high value as data sources for different forecasting methods [3, 4].

To use weather radars to their full potential, one has to have understanding of the physics behind the measurements. It is especially important to acknowledge, how the radar emitted rays are refracted in the atmosphere to understand their propagation and how the power of the ray is attenuated in the atmosphere. Another detail that has to be taken into account is that some radar measurements might not be weather related, for example, a radar beam might propagate in anomalous way or a flock of birds might fly into the beam. Hence, some observations have to be filtered away and replaced using interpolation. Also, as radar usually measures the atmosphere in a sequence of sweeps defined by a fixed elevation or azimuth angle, the values in between these parts have to be interpolated in order to have a full three-dimensional Cartesian raster from the observations, which for example could be used for benefit of forecasts done with extrapolation of radar echoes. A diagram of the interpolation from a radar volume onto a three-dimensional raster is presented in figure 1 [3].

This work concentrates on the interpolation of radar volumes and tests a new method for interpolation of the radar measurements on to a three-dimensional raster. As radars conduct a full scan of the volume in few minutes, the used analysis methods should be as time efficient as possible. Hence, the new method is enhanced for time efficiency.

Some popular choices for the methods used for interpolation of radar volumes are based on basic models such as nearest neighbor (NN) or linear interpolation [5, 6]. These methods interpolate with assumption that the values measured by radar are exact and the interpolation has no uncertainty. The method presented

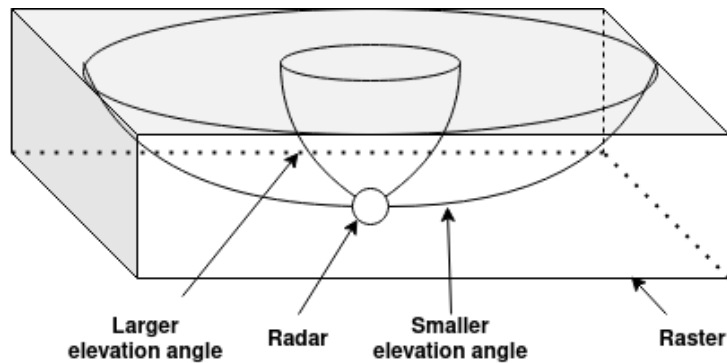


Figure 1: Here a diagram of the interpolation of radar volume onto a three-dimensional raster is presented. Two fixed elevation angles define the two radar measurement sweeps when rotated around the radar. From these measurements, a three-dimensional Cartesian raster denoted by the block in the diagram will be interpolated.

in this work assumes some uncertainty in the measurements and applies Bayesian statistical techniques to regularize a fit made on the measurements. More specifically, the method is based on algorithm known as sparse Bayesian learning (SBL). SBL is able to estimate to what extent the model has uncertainty in its outputted values based on the standard deviation of a Gaussian probability model [5, 6, 7, 8].

SBL was selected for inspection as it would possibly lead to better interpolations through the regularization that the method provides when compared to other methods like NN. The uncertainty estimates obtained via the standard deviation in the Gaussian model could also benefit the interpretability and reliability of the mean interpolation results.

The method is first formulated in an attempt to provide an efficient computational framework, and then it is tested for interpolation of the radar volumes in comparison to some efficient and common methods from a popular radar analysis library wradlib [9]. To briefly describe the framework, a basis for interpolation is defined by type-II discrete cosine transform and a subroutine of matrix inversion using conjugate gradient method (CG) is used to compute an estimate for the posterior mean and the mean interpolation. On top of these, the diagonal of an inverse matrix is computed using CG and a matrix probing approach to obtain variances for the model at each location. These subroutines enable the efficient computational framework [8, 10, 11].

After the method has been introduced, an approach for approximation of an objective function of SBL is provided and demonstrated as a possible way for testing the behavior of the method [12].

SBL is tested using a single radar volume from which the test and training datasets are further divided. Before interpolation this data is corrected for attenuation and non-meteorological echoes are filtered out [13, 14].

The rest of the work is structured into following sections: theoretical background and earlier research, research material and methods, results, discussion, and summary. The first of these sections, section 2, concentrates on theoretical background and earlier research. It consists of an introduction to the weather radar measurements, a

discussion on earlier research on interpolation of weather radar data, and a theoretical background on Bayesian data analysis. This section will serve as a theoretical basis for the rest of the work. The section 3, research material and methods, builds upon the context and theory provided by the previous section, section 2, as the dataset of radar measurements, its preprocessing, as well as SBL built upon Bayesian techniques are presented in detail along with the experimental setup, where SBL will be tested. The experiments comprise of three different runs of SBL compared to methods from wradlib [9]. The results from the experiments are presented in section 4. The second to last section, section 5, offers discussion on the results as well as ideas for further research, and the work concludes with a summary of the thesis.

2 Theoretical background and earlier research

The contributions of this work are mostly based on applications of weather radar technology, Bayesian data analysis, and interpolation. In this section these subjects are presented as a preliminary for the rest of the work and to give insight on the earlier research on these topics.

First, the basics of radar sensing are introduced with a description of radar systems, relevant physics and relevant quantities derived from radar measurements. The mentioned quantities include the input data to be interpolated and the quantities needed for preprocessing the input data. Then insight on earlier research on interpolation of weather radar data as well as on some basic interpolation methods is provided, and to conclude this section, a primer on Bayesian data analysis is presented to provide necessary understanding for the methods introduced later in this work.

2.1 Weather radars

A weather radar is an instrument capable of sensing distant objects using microwaves. The instrument consists of several parts as can be seen in figure 2. First the antenna, located inside a radome for cover, sends a high intensity microwave into a direction defined by the azimuth and elevation angles. This wave is reflected back by scatterers like drops of water, and the echo is collected by the antenna [3].

The antenna obtains the waves from a transmitter and sends the echoes to a receiver through a waveguide. This waveguide passes through a circulator switch which determines whether the radar is transmitting or collecting microwaves. The transmitter creates the microwaves to the waveguide and the receiver collects the echoes from the waveguide. The echoes from the receiver are subsequently postprocessed to produce radar products [3].

2.1.1 Basic measurements

In order for the radar to be useful, it should be able to detect the range of scatterers, as combining range with the azimuth, and elevation angles defines the location of a possible echoing scatterer. The range r_i from the radar to the echoing scatterer is computed by comparing the time difference between the transmission time t_0 and return of the echo at time t_i with the speed of light c through formula

$$r_i = \frac{c(t_0 - t_i)}{2n}, \quad (1)$$

where n is the refractive index [3].

A weather radar captures the power of the echo from the scatterers in the atmosphere and computes an estimate for a quantity called radar reflectivity factor per unit volume. Using this quantity, one can determine that at a given range there are hydrometeors, since if there is no or a very faint echo at given range, the range should not contain significant hydrometeors. The quantity is also known as

reflectivity, and it is measured in the units mm^6/m^3 . It is denoted as Z and defined as

$$Z = \int_0^\infty N(D)D^6dD, \quad (2)$$

where D is the diameter of hydrometeors in millimeters and $N(D)$ is the function number of hydrometeors per unit volume in units m^{-3} [3]. Reflectivity is usually presented in logarithmic units dBZ. This can be done by a logarithmic conversion as

$$Z_{\text{dBZ}} = 10 \log_{10}(Z/(1 \text{ mm}^6/\text{m}^3)) \quad (3)$$

[3].

As the properties of scatterers from which the radar receives echoes are not usually known a priori, the radar system can not be used to measure reflectivity itself. Hence, a weather radar outputs a quantity called equivalent reflectivity Z_e . The quantity is computed from the power of the received signal with the assumption that the scatterers are water drops. In order to obtain the equivalent reflectivity, in [3] a modified Probert-Jones radar equation is given as

$$P_r = \frac{1.22^2 0.55^2 10^{-18} \pi^7 c P_t \tau D_a^2 T(0, r)^2}{1024 \log_e(2) \lambda^4 r^2} ||K_w||^2 Z_e, \quad (4)$$

where P_r and P_t are the power of the reflected and the transmitted signal respectively, c is the speed of light, τ is the duration of the transmitted pulse, D_a is the diameter of the antenna, λ is the wavelength, which is about 5 cm for a C-band radar, $T(0, r)$ is the transmittance of the atmosphere along the path from the radar to the scatterer at range r , and $||K_w||^2 = 0.93$ is the dielectric constant of liquid water. The numeric coefficients and exponents result from modifications and approximations for Rayleigh scatterers [3, 15].

Equivalent reflectivity is given in the same units as reflectivity, and it describes the reflectivity itself when the scatterers are water. According to [3], rain with rainrate of about 1 mm/h has equivalent reflectivity of about 25 dBZ, heavy rain from convective shower with rain rate of 20 mm/h has equivalent reflectivity of about 45 dBZ, while hail or a thunderstorm with rain rate of over 100 mm/h might have equivalent reflectivity of 55 dBZ [3].

According to [16] there exists some evidence that rainfall rates R are lognormally distributed, that is, $R \sim e^X$, where $X \sim \mathcal{N}(\mu, \sigma)$. Hence, as [3] states that $Z = aR^b$, where Z is in terms of reflectivity and a and b are constants, it must hold that Z in dBZ is normally distributed given that there is rain. Hence, a radar volume possibly contains normally distributed regions [16].

On top of the power from the returned radar signal, the phase of the return is also useful for some quantities. The phase of the radar return φ_i can be computed as

$$\varphi_i = 2\pi f(t_0 - t_i), \quad (5)$$

where f is the frequency of the transmitted wave, t_0 is the transmission time, and t_i is the time of the radar return [3].

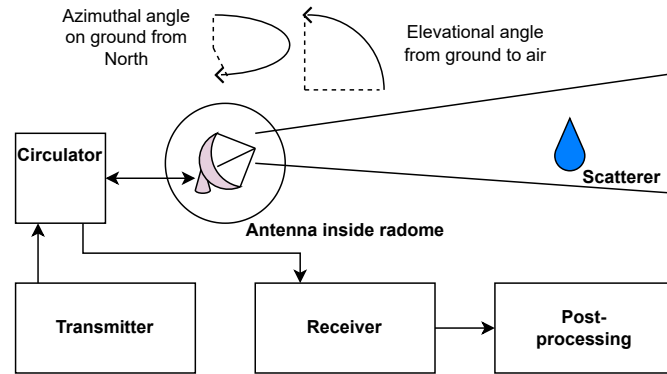


Figure 2: This figure presents a high level description of components of a radar system. The transmitter feeds a generated signal to the circulator which switches the radar to receiving or transmitting mode. If the radar is transmitting, the waveguide leads the signal to the antenna which sends the beam towards scatterers in the atmosphere. The antenna is located inside a radome that covers the radar. The circulator will switch to receiving mode in order to collect the echoes from possible scatterers noticed by the previous beam. The echoes from the antenna are led through the waveguide to the receiver where they are detected. The received signals are post-processed to produce radar products such as the reflectivity PPI from a certain elevation angle. The azimuth and elevation angles define the direction to which the antenna is controlled to point at any given time [3].

A single ray produces a sequence of radar bins the length of which in radial dimension is defined by the distance that the pulse travels during its duration. Radar sweeps are then sets of these sequences of bins produced when radar is rotated either with respect to the azimuth or the elevation angle. This essentially defines a sector of a circle around the radar. Different types of sweeps are used to create a radar volume which is defined as a collection of sweeps. The formation of these sweeps during a full scan of the volume is specified by different scanning strategies that might vary by weather event [17, 18].

Radar measurements can be presented in different ways, for example, by a plan position indicator (PPI) and a range-height indicator (RHI). PPIs are created by rotating the radar with respect to the azimuth angle but having a fixed elevation angle, while RHI are created by rotating the elevation angle but having a fixed azimuth angle. After data from scans is modified, its presentation is known as a radar product. One of these products, in addition to PPI or RHI, is a constant altitude plan position indicator (CAPPI). CAPPIs essentially visualise radar data at constant altitude in each direction defined by an azimuth angle [3].

2.1.2 Dual-polarization measurements

Modern radars are also capable of making measurements of the shape of hydrometeors with the dual-polarization functionality. Dual-polarization refers to the capability of receiving and transmitting vertically or horizontally oscillating electromagnetic

radiation. In practice, operational radars can transmit a combination of these simultaneously, and subsequently measure the polarized components of the received echo. The echo is then measured for horizontal and for vertical oscillation. The measured scatterers might have different size in horizontal and vertical dimensions, and hence the obtained echo might have different in properties for its polarizations. Consequently, the difference in reflectivity measured from the different polarizations of the echoes is related to the shape of the scatterers. The quantity Z_{DR} , known as differential reflectivity in units of dB, is used to describe the shape, and it is defined as

$$Z_{DR} = 10 \log_{10} \left(\frac{Z_h}{Z_v} \right), \quad (6)$$

where Z_h and Z_v are the reflectivity values from horizontally and vertically polarized components of the echo in linear units [3]. Different scatterers have their own ranges of differential reflectivity values, for example, rain can have Z_{DR} of 0 dB to 5 dB, where the greater value means a larger drop size, and for ice crystals different values correspond to different shapes, like needles at about 1.5 dB and plates at 4 dB [3, 19].

Another relevant parameter is cross correlation ratio, also known as copolar correlation coefficient $\rho_{co}(r)$. Cross correlation ratio is computed as the correlation of the two orthogonally polarized signals at a given range interval the length of which is specified by the duration of the wave pulse sent by the radar and the speed of the electromagnetic wave. Then $\rho_{co}(r)$ for a given range r is specified as

$$\rho_{co}(r) = \frac{|\text{cov}_{V,H}(r)|}{\sqrt{\text{cov}_{H,H}(r)\text{cov}_{V,V}(r)}}, \quad (7)$$

where the covariances are computed as

$$\text{cov}_{V,H}(r) = \frac{1}{M_s(r)} \sum_{j \in S(r)} |V_j H_j| \exp(i(\varphi_{H_j} - \varphi_{V_j})), \quad (8)$$

where H_j and V_j are the powers of the sampled signals at sample j , φ_{H_j} and φ_{V_j} are the phases of the radar returns for the respective polarizations, $S(r)$ is the set containing indices of the sampled signal corresponding to the signal at the range interval to which r belongs, and $M_s(r)$ is the size of the set $S(r)$. As an example, rain has ρ_{co} greater than 0.9, while biological scatterers tend to have ρ_{co} smaller than 0.9. Different types of hail and snow typically have ρ_{co} greater than 0.75 [3].

In practise, these dual-polarization quantities together with information on the melting layer, obtained, for example, from a sounding or an NWP, can be used to classify radar echoes. The melting layer is the area at certain height at the troposphere where icy particles melt. The specification of the altitude of the layer is important as otherwise the amount of rainfall would be overestimated due to high reflectivity values given by the layer [3, 14].

2.1.3 Attenuation

Before classifying the radar echoes, it is beneficial to do attenuation correction on the measurements. Attenuation is anything that makes microwaves lose power. Its effect

on the microwaves depends, for example, on the temperature, the type of traversed medium, and the wavelength of the radiation. Different polarizations are also attenuated differently, and hence, in addition to reflectivity, differential reflectivity is attenuated too. One can apply correction algorithms to counter attenuation, for example, [13] provides such an algorithm [3, 13, 19].

The differential phase shift ψ_{dp} is computed from the phase difference of the vertically and horizontally polarized components of the returned echo. More specifically, the differential phase shift can be computed as

$$\psi_{dp} = \varphi_{HH} - \varphi_{VV} = \delta_{co} + \Phi_{dp} + \psi_o, \quad (9)$$

here φ_{HH} and φ_{VV} are the phases of horizontally and vertically polarized components of the echo, respectively, δ_{co} is the differential back scattering phase delay, Φ_{dp} is the two-way differential propagation phase, also known as just differential phase, and ψ_o is the phase difference between the orthogonally polarized transmit waves at range zero [3]. From the two-way differential propagation phase, one can compute its rate of increase with respect to the range known as the specific differential phase delay K_{dp} in units degree per km. The specific differential phase delay is a useful quantity for the estimation of rainrates as it not affected by attenuation [3].

2.1.4 Refraction

On top of attenuation, one sometimes needs to consider refraction in the atmosphere for sensible radar measurements. For example, to produce CAPPIs correctly, one needs to have understanding of the correct altitude of the radar bin. The rays emitted by the radar propagate in curved arcs as the atmosphere has a non-uniform refractive index. Also, taking into account that the Earth is curved, the height h of a scatterer and the distance on the surface of the Earth s can be approximated in the troposphere as

$$h = \sqrt{r^2 + (k_e a)^2 + 2rk_e a \sin \theta_e} - k_e a, \quad (10)$$

and

$$s = k_e \arcsin \frac{r \cos \theta_e}{k_e a + h}, \quad (11)$$

where r is the measured range, θ_e is the measured elevation angle of the radar, k_e is a constant, and a is the radius of the Earth. The constant is usually set to $k_e = \frac{4}{3}$ since it approximates the values well in the troposphere for most elevation angles and weather radar wavelengths. More specifically, it is assumed that the refractive index follows the exponential law $n(h) = n_s e^{-0.1439h}$, where n_s is the refractive index on the surface. The propagation of the rays with different elevation angles is presented in figure 3 [20].

2.2 Interpolation of weather radar data

In numerical analysis interpolation is often done using exact methods that reproduce the known function values exactly. Exact methods in the context of geostatistics

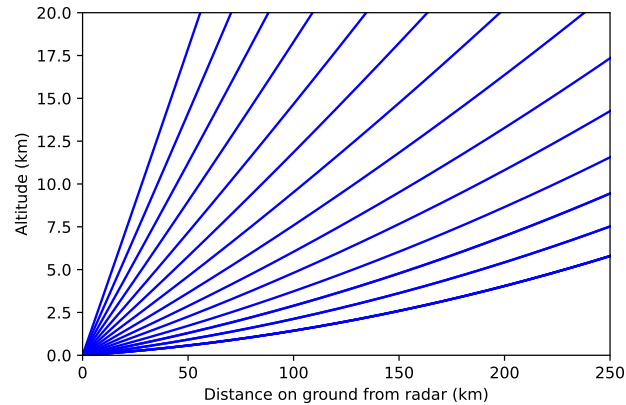


Figure 3: A plot of the propagation of the rays based on the equations (10) and (11). The elevation angles are from lowest to highest 0.5, 0.9, 1.3, 2.4, 3.1, 4.0, 5.1, 6.4, 8.0, 12.5, 15.6, and 19.5 degrees as in VCP 212 scanning strategy for NEXRAD radars discussed in section 3.1.

could be, for example, splines or linear interpolation. In geostatistics, interpolation can also often be inexact, so that the interpolation result, also known as interpolant, does not pass exactly through the known values. Interpolation in geostatistics could be then defined as modeling of values inside the domain that the observations cover, when the observations are used for the model through some method. Interpolation methods in geosciences could be further divided into two classes: deterministic and non-deterministic. Deterministic methods assume that the obtained measurements do not contain randomness, and they usually are methods that assume the values at the measured locations are exact. Non-deterministic methods assume that some level of uncertainty exists, but depending on the method the values can be treated as exact at measured locations. NN and inverse distance weighting (IDW) are examples of deterministic interpolation methods, while estimation by multivariate linear regression model is an example of a non-deterministic interpolation method. This is exemplified in figure 4 [21]. In [21] several deterministic, such as kernel approximations and splines, as well as non-deterministic interpolation methods, such as radial basis functions, are presented for spatial interpolation.

As radar measurement are tied to a location, spatial interpolation, or interpolation of data on a spatial domain, is needed in different applications dealing with radar data. When clutter is removed from radar measurements, the now emptied locations need to be refilled by interpolation. In order to enhance the resolution of the grid containing the radar measurements, the values between the available measurements can be obtained by interpolation. Interpolation is also applied to form PPIs, RHIs, and three-dimensional rasters from measurements of a radar or to composites containing data over the reach of the measurements from several radars [6, 22, 23].

The aforementioned examples deal with interpolation of the reflectivity, and in [6] it is discussed whether one should do linear interpolation with respect to the rays, elevation, and azimuth in linear or logarithmic units of reflectivity. Linear units

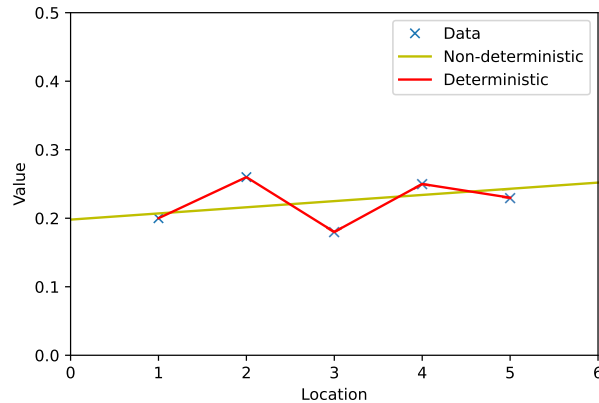


Figure 4: Here some example data, a linear regression fit to the data, and a linear interpolation on the data are presented. Linear regression is an example of a non-deterministic interpolation method, and linear interpolation is a deterministic interpolation method. In this example, the data is used to produce a regression fit to the noise so that the values are not exact at the measured locations. Interpolation would then be done by reading the values from the plots between the interval specified by the first and the last locations [21].

seem to produce a smaller error in general for the used dataset, but the results also seem to vary depending on the magnitude of the reflectivity. The same problem was inspected in [24], where the results were contradictory as logarithmic reflectivity was preferred for interpolation [6, 24].

In addition to interpolation of reflectivity, other radar measured quantities have also been interpolated in earlier research. For example, wind velocity data obtained from radar measurements [25, 26] and [27].

As radar finishes a measurement of a sweep in the order of ten seconds, it takes few minutes to measure the whole volume containing several sweeps. Measurements are not simultaneous, and hence interpolation in temporal dimension becomes relevant as well. Methods that interpolate in temporal dimension on top of the spatial dimensions are presented in [28], where a nowcasting scheme was assessed for temporal interpolation, in [27], where some data was inpainted with respect to the time dimension, and in [29] where optical flow was applied.

2.2.1 Methods in earlier research

In several research articles related to interpolation of radar data, the methods are based on models such as selecting the NN or applying linear interpolation [6, 23]. These are deterministic interpolation methods that have low computational complexity. To counter performance issues created by higher time complexity of a method, such as Ordinary Kriging (OK) in [22], interpolation is made only in vertical or horizontal cross-sections and only for data nearby the point which is to be modeled by interpolation. The nearby data is obtained using a nearest neighbors

approach. These restrictions to input data allow for more sophisticated computational methods to be used as there are drastically less data in this approach than in a three-dimensional raster without nearest neighbors approximation [6, 22, 23].

Other common tools for interpolation of meteorological data to a grid are Barnes and Cressman interpolation schemes. These methods are used, for example, to map radar measurements from polar coordinates to Cartesian coordinates on PPIs [30, 31].

In the formerly discussed research, the methods for interpolation of radar data between elevation angles are based on simple models but there exists models that are more sophisticated. For example, in [5] a mosaic is created by interpolating data from several radars through a decision tree approach for selection of the correct value for each interpolated location. In [26] weather radar PPIs were interpolated for super-resolution using sparse representation constructed from similar patches from the image at other locations. Then in [27] a convolutional neural network was used to interpolate or inpaint areas of missing data.

Research on spatial interpolation methods for multidimensional data exist on other subfields of geoscience, for example, in [32] where variational sparse Bayesian learning was used to interpolate geo-data in three dimensions and in [33] where dictionary learning was used to reconstruct seismic data in high-dimensional dataset.

2.2.2 Metrics in earlier research

In [5, 6, 22, 24], and [29] the reliability of interpolation results are evaluated in terms of the root mean squared error metric (RMSE). Furthermore, RMSE values are compared with rainfall rates derived from the interpolated raster with respect to the observed rainfall rates in [22, 28, 29], but also more direct comparison to explicit radar observed values exist in [6] and [26]. For RMSE, the larger the absolute error of a point is, the more weight it has on the error. Hence, RMSE is more sensitive to outliers, which means that larger errors matter more in the metric when compared to other errors. The mean absolute error (MAE) is more robust to outliers. The distribution of errors have also an effect on choosing the metrics as MAE benefits from uniform distribution while RMSE has advantages with normal distribution [34].

Other metrics used in literature are bias in [5, 6, 29], power spectral density as well as radially averaged power spectral density in [26] and [28], respectively. In order to have focus on the speed of the interpolation for practical applications, CPU time was also taken as one evaluation metric in [28]. Qualitative approaches to evaluate the interpolation based on observing anomalies from a visualization of radar measurements, are presented in [5] and [23].

2.3 Basics of Bayesian data analysis

Bayesian data analysis is based on the application of the Bayesian theorem

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}, \quad (12)$$

where A and B are some events, and the conditional probability is denoted with $p(B|A)$ which essentially describes the probability of event B given that event A

happens. Furthermore, conditional probability is defined using the fraction

$$p(B|A) = \frac{p(B \cap A)}{p(A)}, \quad (13)$$

where $B \cap A$ is the intersection of the two events [35].

There might exist an idea of what the probability of A should be, for example, from another study. This probability is denoted in the Bayes' theorem (12) as $p(A)$ which is also known as the prior probability of A . The probability $p(B|A)$ is called likelihood. Likelihood could be defined by some statistical model that gives probability for the event B given event A , which corresponds to a parameter in the model having a certain value. Likelihood is essentially the same statistical model that creates estimates from data in classical statistics with the parameter A given as part of the model. Combining the likelihood and the prior will then result in a posterior probability $p(A|B)$, which presents the updated information for the probability of A given that event B happens. The only term unexplained in (12) is $p(B)$ which is known as the marginal likelihood or evidence. This probability can be computed by summing or integrating over the values of A in the joint probability of A and B in the numerator of equation (12) [35].

Bayesian data analysis works on two levels of inference. The first one is model fitting and the second is model comparison. During model fitting, the statistical model is fitted almost the same way as in classical statistics with the exception of application of prior information. In terms of probabilities, this would correspond to computation of the posterior distribution using the Bayes' theorem (12), where the model parameters \mathbf{w} are inferred given the data D and the model \mathcal{H}_i as

$$p(\mathbf{w}|D, \mathcal{H}_i) = \frac{p(D|\mathbf{w}, \mathcal{H}_i)p(\mathbf{w}|\mathcal{H}_i)}{p(D|\mathcal{H}_i)} \quad (14)$$

[36].

After the first level of inference, one performs model comparison, during which one tries to estimate probabilities for different models as

$$p(\mathcal{H}_i|D) \propto p(D|\mathcal{H}_i)p(\mathcal{H}_i), \quad (15)$$

where one can assign a prior $p(\mathcal{H}_i)$ for different models [36]. The higher the probability for the model given the data is, the better it describes the data. If the prior for the different models is a uniform distribution, that is, $p(\mathcal{H}_i)$ is constant, the model comparison turns into a problem of maximizing the evidence $p(D|\mathcal{H}_i)$ in proportionality (15). This approach for model comparison is possibly the most significant benefit of Bayesian data analysis as classical data analysis lacks as simple and widely applicable framework for selecting the best model that is described by the data [36].

The prior distributions for the parameters in priors are called hyperpriors. Hyperpriors are defined similarly as normal priors. For computational convenience and for interpretability, it might make sense to define priors by conjugate prior distributions.

Definition 2.1 (Conjugate prior [35]). If \mathcal{F} is a class of sampling distributions $p(y|\theta)$, and \mathcal{P} is a class of prior distributions for θ , then the class \mathcal{P} is *conjugate* for \mathcal{F} if

$$p(\theta|y) \in \mathcal{P} \text{ for all } p(\cdot|\theta) \text{ and } p(\cdot) \in \mathcal{P}.$$

For example, the gamma distribution is a conjugate prior for the inverse of the variance, i.e., precision, of normal distribution [35].

To learn a model in a Bayesian way, one can apply the expectation-maximization (EM) algorithm 1, given a set of latent variables γ with some known form of distribution with unknown parameters. Essentially, the model learns a local maximum for the marginal posterior distribution $p(\phi|y)$, where ϕ is the some value that is modelled and y is the data sample. This corresponds to the maximization of the left hand side in the model comparison proportionality (15). The algorithm consists of iteration of two alternating steps: the expectation step and the maximization step. In the expectation step, presented from row 3 to row 6 of algorithm 1, the expectation of the model with respect to the current parameter estimates of a latent variable, which is part of the model, is determined. Then in the maximization step, presented on the seventh and the eighth lines of algorithm 1, the obtained expression for the expectation is maximized. EM is useful if the expectation can be expressed as an analytic function of the parameter that is maximized during the algorithm given the distributions [35].

Algorithm 1 EM algorithm [35]

Input:

Data y , initial value parameter ϕ^{new} , initial estimate γ^{new} , convergence tolerance ε

Output:

The final value for the parameter ϕ^{new} and the final value for γ^{new}

1: **do**

2: Assign the value of ϕ^{new} to ϕ^{old} and γ^{new} to γ^{old}

3: **E-step:** Determine the expected log posterior density function from,

4:

$$\mathbb{E}_{\text{old}}(\log p(\gamma, \phi|y)) = \int p(\gamma|\phi^{\text{old}}, y) \log p(\gamma, \phi|y) d\gamma,$$

5: where the expectation is taken over the conditional posterior distribution
6: of γ , given current estimate for ϕ^{old} .

7: **M-step:** Assign ϕ that maximizes $\mathbb{E}_{\text{old}}(\log p(\gamma, \phi|y))$ to ϕ^{new}

8: Assign the mean values for γ given parameter value ϕ^{new} to γ^{new}

9: **while** $\|\gamma^{\text{old}} - \gamma^{\text{new}}\| < \varepsilon$

Return: ϕ^{new} and γ^{new}

3 Research material and methods

In this section the research material and methods used for interpolation are presented. First the dataset is introduced along with methods to preprocess it. Then the formulation of SBL is presented in detail. The section ends with description of the experimental setup, where SBL and other methods are tested using the dataset.

3.1 Dataset and preprocessing

The dataset consists of data extracted from NEXRAD WSR-88D radar at Fort Worth, Texas. The specific location of the radar next to Dallas on OpenStreetMap is presented in figure 5, and it can be seen that the radar measures precipitation on an area with population of over 7 million people according to 2020-2021 population estimates in [37]. Hence, the precipitation measurements from this radar are of high importance [38, 39, 40].

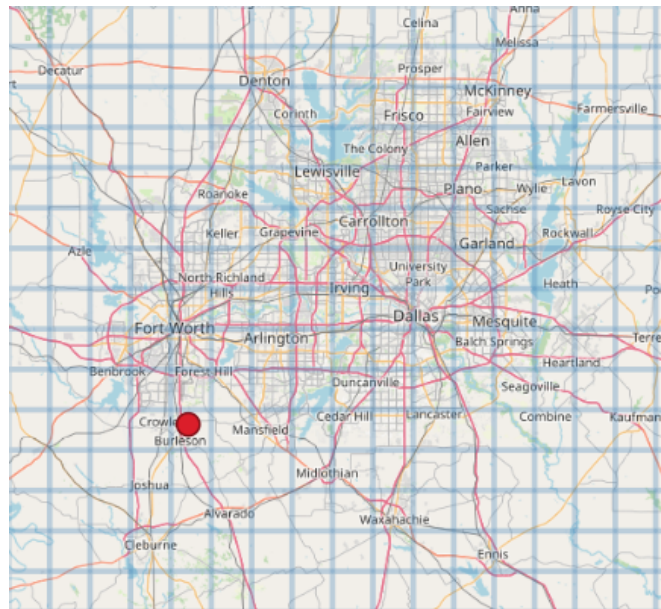


Figure 5: The location of the NEXRAD WSR-88D radar at Fort Worth, Texas is represented with a red point. One grid cell on the background is of 10 km in size in the north-south and the east-west directions. There are about 7 million people living in the area, and hence the measurements of the radar are of high importance. The map is from OpenStreetMap [40].

For simplicity and due to time restraints, only one radar image is used for testing. The timestamp of the volume is 9th of June 2019 at 19.03:20 UTC, which essentially describes the time when the scanning of the whole volume was started. Even though the radar measurements are not simultaneous, it is assumed that all values in the radar volume are measured simultaneously to allow a simple approach for the interpolation of the radar volume. The weather event in the volume is an example of a mesoscale convective systems (MCS), which essentially means that they contain convective and

stratiform precipitation. This translates to severe weather on large areas and can be visible on radar measurements with high reflectivity values on a broad area [41, 3].

The whole volume scan takes about 4.9 minutes, and the radar scans are conducted using VCP 212 scanning strategy with a Supplemental Adaptive Intra-Volume Low-Level Scan (SAILS), which means that the whole volume contained 19 scans at 14 elevation angles where the lowest elevation is scanned four times, the second as well as third lowest are scanned twice, and the rest are scanned only once. The usage of these scans for the experiments is discussed later in the description of the experimental setup in section 3.3. The resolution of the radar bins in azimuth angles is 0.5 degrees for the sweeps with three lowest elevation angles and 1 degrees for the rest of the sweeps. The resolution in radial direction is 250 meters. Every sweep does a single full rotation during a scan to produce a PPI. An example of the PPI plots, where no corrections to reflectivities are made, is shown in figure 6 [18, 39].

These scans are attenuation corrected and the echoes are classified as discussed later in sections 3.1.1 and 3.1.2. After classification bins with clutter, anomalous propagation as well as biological scatterers are removed. For attenuation correction and echo classification the melting layer is detected from rawinsonde temperature sounding from Forth Worth, Texas available through rawinsounding archive Iowa Environmental Mesonet (IEM RAOB), and is set to equal 4800 m above the sea level measurement at midnight on 10th of June 2019 UTC [13, 14, 42].

To produce interpolation on a region where there are enough nearby measurements, the area used for interpolation is filtered with respect to the distance of the bin at the sea level of the Earth and the altitude of the bin using the formulas (10) and (11). The values with larger than 80 km distance from the radar in the horizontal Cartesian x or y coordinates are filtered away as well as the values further than 6.4 km in the vertical z direction in the coordinates from the antenna.

The dataset is split into test and training datasets based on the elevation angles. The splits for different test cases are presented in figure 11. The details of this split and the test cases are discussed in detail in section 3.3.

3.1.1 Attenuation correction

Reflectivity and differential reflectivity are both affected by attenuation and hence both are corrected. The correction is made using the method described in [13] with a small modification exponent for the correction of differential reflectivity. In the method, the specific attenuation A_h is estimated using the expression

$$A_h(r) = \frac{[Z_a]^b [10^{0.1b\alpha\Delta\Phi_{DP}(r_0;r_m)} - 1]}{I(r_0;r_m) + [10^{0.1b\alpha\Delta\Phi_{DP}(r_0;r_m)} - 1]I(r;r_m)}, \quad (16)$$

where Z_a is the radar measured attenuated reflectivity in linear units, α and b are pre-determined constants, $\Delta\Phi_{DP}(r_0;r_m)$, is the difference between the differential phase at ranges r_0 and r_m , where r_0 is the range at the beginning of the echo and r_m is the range of the beginning of the melting layer, and $I(r_0;r_m)$ as well as $I(r;r_m)$ are defined by

$$I(y;z) = 0.46b \int_y^z [Z_a(s)]^b ds \quad (17)$$

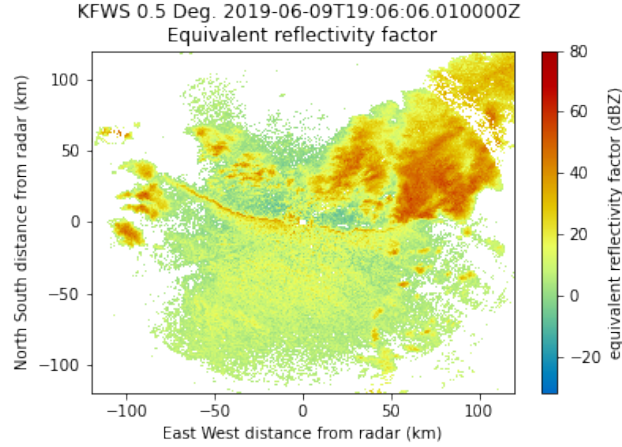


Figure 6: A raw PPI plot for the lowest elevation angle in the dataset. The plot shows the computed reflectivity at gates, where the radar obtained a returned echo in dBZ. As the dataset follows the VCP 212 scanning strategy, the volume contains several scans of the lowest elevation angles. This PPI plot is restricted to visualize only about 120 km in both North-South and East-West directions. The maximum distance that a weather radar can measure in practice can be several hundreds of kilometers.

[13].

Given the specific attenuation, the path integrated attenuation or PIA at range r_m can be computed as

$$\text{PIA}(r_m) = 2 \int_{r_0}^{r_m} A_h(s) ds, \quad (18)$$

and it can then be summed without modifications with the attenuated reflectivity expressed in dBZ units to obtain the attenuation corrected reflectivity in dBZ [13].

The corrected differential reflectivity can then be computed using a similar formula but with different constants. Specific differential attenuation A_{DP} is computed with the specific attenuation modified with pre-determined multiplicative as well as exponential constants γ and d , respectively, with modified version of the formula given as

$$A_{DP}(r) = \gamma A_h(r)^d \quad (19)$$

[13]. Then given the specific differential attenuation, the corrected differential reflectivity $Z_{dr}(r)$ at range r can be obtained from the radar measured attenuated differential reflectivity $Z_{dr}^{(a)}(r)$ as

$$Z_{dr}(r) = Z_{dr}^{(a)}(r) + 2 \int_{r_0}^r A_{DP}(s) ds, \quad (20)$$

here r_0 is still the range at the beginning of the echo [13].

The used values for the pre-determined coefficients α, b, γ, d in equations (16) and (19) are given as $\alpha = 0.08, b = 0.64884, \gamma = 0.3, d = 1.0804$. These terms

are incorrectly valid for a C-band radars with wavelength of about 5 cm, while the WSR-88D radar at Fort Worth, Texas is in fact a S-band radar with wavelength of about 10 cm [13, 43].

The altitude of melting layer is used as a threshold above of which the reflectivity and differential reflectivity values are not corrected. This is due to the fact that the attenuation properties are different for different hydrometeors. These attenuation formulas are for heavy rain, and for example, the same formulas would not work for hail or in the melting layer [13, 43].

3.1.2 Echo classification

Echo classification is done to remove clutter, anomalous propagation and biological scatterers from the radar volume. A simplified version of the used hydrometeor classification algorithm (HCA) is presented in [14]. The algorithm uses the principles of fuzzy logic to distinguish between ten echo classes. The first two classes are ground clutter as well as anomalous propagation (GC/AP) and biological scatters (BS). These classes are filtered out before interpolation, while the remaining eight hydrometeor-related classes are considered in the interpolation [14].

More specifically, the algorithm takes as input the attenuation corrected reflectivity and the corrected differential reflectivity as well as the correlation coefficient. The specific differential phase and the texture parameter of differential phase could also be considered but their usage was omitted in this work. In addition to these parameters, an estimate for the texture parameter $SD(Z)$ of the corrected reflectivity is computed by radially averaging the reflectivity data using a 1-km running average window. The corrected reflectivity, the corrected differential reflectivity, and the correlation coefficient are averaged through the radial direction using a 1-km window for the reflectivity and a 2-km window for the differential reflectivity and correlation coefficient [14].

To classify the echoes, the algorithm uses a membership functions vector to determine how each parameter value affects the likelihood of a certain class, and a confidence vector to weight the significance of the membership function. The membership functions $P^{(i)}(V_j)$, where i corresponds to the echo class and j corresponds to the value of a given input parameter, are specified by a trapezoidal presented in figure 7. The locations x_1, x_2, x_3 , and x_4 are either pre-determined constants or functions of the reflectivity [14].

The confidence vector \mathbf{Q} introduced in [14] describes the level of confidence that each measurement of input data is correct. The echo classification implemented in this work sets the confidence to ones for simplicity, and the confidence of data is therefore not taken into account [14].

The above introduced input parameters are used to compute aggregation values A_i for each class i , to classify the echoes as

$$A_i = \frac{\sum_{j=1}^6 W_{ij} P^{(i)}(V_j)}{\sum_{j=1}^6 W_{ij}}, \quad (21)$$

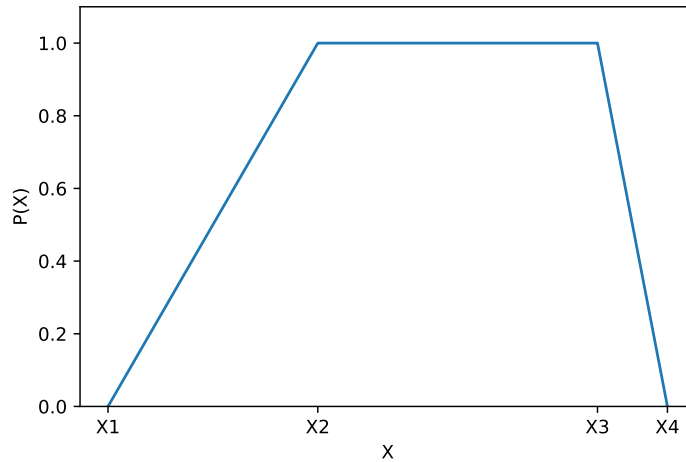


Figure 7: Here the form of the trapezoidal membership functions $P^{(i)}(V_j)$, where i corresponds to the echo class and j corresponds to the value of a given input parameter, is presented for fuzzy logic in the hydrometeor classification algorithm in terms of the predetermined locations x_1, x_2, x_3 and x_4 [14].

where W_{ij} is a pre-fixed weight for the effect of input variable j on class i [14].

As a final step of the HCA algorithm, the highest aggregation value is chosen as the class of the echo given that the chosen class satisfies a hard threshold specified for each echo class [14].

3.2 Sparse Bayesian learning

SBL is a method for solving least-squares problems with regularization. A least-squares problem is solved when the sum of squared residuals between the data and a fit is minimized. On a high level, SBL is an inexact non-deterministic interpolation method, so it fits to noisy data similarly as linear regression in figure 4. Regularization, on the other hand, means that on top of least squares fitting, the degrees of freedom of the model are reduced to avoid spurious overfitting.

The regularization is learnt in SBL by implicitly maximizing the evidence for regularization parameters, which corresponds to Bayesian model comparison. This formulation gives an advantage over classical interpolation methods as it provides a natural probabilistical way to counter over-fitting. Similarly to kriging, SBL provides estimates for the uncertainty of the interpolation [7, 32, 36, 44].

In this section, first a mathematical description of SBL is provided. Then the method is enhanced for efficiency by using a basis that makes the computation task time- and memory-efficient. Lastly, as SBL is based on an iteration that maximizes its objective function, a discussion for an approach to approximate the objective function is provided with an aim to assess the convergence. Next the problem and the model are presented mathematically in [7] and [8].

3.2.1 Model description

Let $y(\mathbf{x})$ denote the value of a spatially varying function at the position $\mathbf{x} \in \Omega$, and let $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_N(\mathbf{x}))^T$ be a vector of N basis functions supported in Ω . After sampling these vectors at the positions of interest \mathbf{x} , the formed matrix is denoted as Φ , where columns and rows are varied by the basis vector and the sampled location, respectively. The function y is then modeled as a weighted linear combination of the basis functions expressed formally as

$$y(\mathbf{x}; \mathbf{w}) = \sum_{i=1}^N w_i \phi_i(\mathbf{x}), \quad (22)$$

where $\mathbf{w} = (w_1, w_2, \dots, w_N)^T$ is a vector of weights for the basis functions [7].

Let $\{\mathbf{x}_n, t_n\}_{n=1}^M$ be a set of sampling locations \mathbf{x}_n and corresponding independent samples t_n . These are modeled with equation (22) and Gaussian noise as

$$t_n = y(\mathbf{x}_n; \mathbf{w}) + \epsilon_n, \quad (23)$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$, and the value for the variance σ^2 will be learnt using the model [7].

Now $\mathbf{t} = (t_1, \dots, t_M)$ is modeled as a Gaussian random variable conditional on the weights \mathbf{w} and variance σ^2 as

$$p(\mathbf{t}|\mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-\frac{M}{2}} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{t} - \Phi\mathbf{w}\|^2\right\}. \quad (24)$$

Equation (24) describes the likelihood for observations given the weights of the basis \mathbf{w} and the variance of the observations σ^2 . In SBL, the priors for the weights \mathbf{w} are defined similarly as the prior for \mathbf{t} , that is, the components of \mathbf{w} are modeled as independent normally distributed variables with mean zero and some variance. More formally, the prior for \mathbf{w} is given as

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=1}^N \mathcal{N}(w_i|0, \alpha_i^{-1}), \quad (25)$$

where the vector $\boldsymbol{\alpha}^{-1} = (\alpha_1^{-1}, \dots, \alpha_N^{-1})$ defines the variance for the prior weights [7].

The precisions for normal distributions have conjugate priors defined by gamma distributions [35]. Hence the precisions $\boldsymbol{\alpha}$ and $\beta = \sigma^{-2}$ can be efficiently modelled with gamma distributed hyperpriors defined as

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^N \text{Gamma}(\alpha_i|a, b), \quad (26)$$

and

$$p(\beta) = \text{Gamma}(\beta|c, d), \quad (27)$$

where $\text{Gamma}(\alpha|a, b) = \Gamma(a)^{-1} b^a \alpha^{a-1} e^{-b\alpha}$, and $\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt$ [7].

Learning the precisions for the Gaussian prior distributions of the weights \mathbf{w} enables SBL to promote sparsity as for larger precisions the weights are more likely to be zero. This sparsity behavior is made possible by specification of a threshold distribution which is defined to correspond to a weight of value zero. This threshold is fixed before the iteration and its behavior is demonstrated in figure 8 [7].

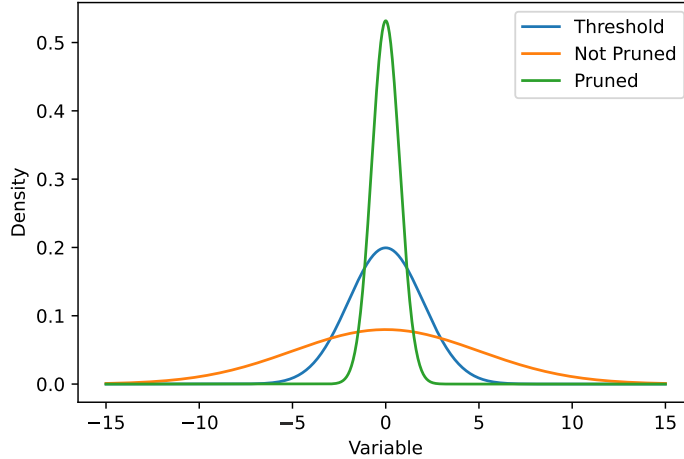


Figure 8: This figure demonstrates how during the pruning procedure it is decided whether a basis function is pruned or not given a threshold. The blue curve corresponds to the density with threshold precision. Any density with more precise value for zero is pruned away. Values which are pruned away are visualized by the green curve concentrated more tightly around zero, while the yellow curve corresponds to a value not pruned away as it is more likely for it to be non-zero [7].

Now that the likelihood and the priors have been defined, the posterior can be defined using Bayes' theorem (12). Using the formula for conditional probability, the posterior over all unknowns $p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2 | \mathbf{t})$ is split for computational convenience as

$$p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2 | \mathbf{t}) = p(\mathbf{w} | \mathbf{t}, \boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{\alpha}, \sigma^2 | \mathbf{t}) \quad (28)$$

[7]. The posterior $p(\mathbf{w} | \mathbf{t}, \boldsymbol{\alpha}, \sigma^2)$ can be evaluated using the Bayes' theorem (12) since $p(\mathbf{t} | \boldsymbol{\alpha}, \sigma^2)$ can be given explicitly using properties of Gaussians. More formally, since

$$\begin{aligned} p(\mathbf{t} | \boldsymbol{\alpha}, \sigma^2) &= \int_{\mathbb{R}^N} p(\mathbf{t} | \mathbf{w}, \sigma^2) p(\mathbf{w} | \boldsymbol{\alpha}) d\mathbf{w} \\ &= (2\pi)^{-N/2} |\sigma^2 \mathbf{I} + \Phi \mathbf{A}^{-1} \Phi^T|^{-1/2} \exp \left\{ -\frac{1}{2} \mathbf{t}^T (\sigma^2 \mathbf{I} + \Phi \mathbf{A}^{-1} \Phi^T)^{-1} \mathbf{t} \right\}, \end{aligned} \quad (29)$$

where both terms inside the integral are Gaussians, and hence the integral can be computed explicitly [7]. Then by [7], the posterior distribution over weights

$p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2)$ is given as

$$\begin{aligned} p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2) &= \frac{p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})}{p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)} \\ &= (2\pi)^{-(N+1)/2} |\boldsymbol{\Sigma}|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{w} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{w} - \boldsymbol{\mu}) \right\}, \end{aligned} \quad (30)$$

where the posterior covariance $\boldsymbol{\Sigma}$ and the posterior mean $\boldsymbol{\mu}$ are defined as

$$\boldsymbol{\Sigma} = (\sigma^{-2} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \mathbf{A})^{-1}, \quad (31)$$

and

$$\boldsymbol{\mu} = \sigma^{-2} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{t}, \quad (32)$$

where $\mathbf{A} = \text{diag}(\alpha_1, \dots, \alpha_N)$ [7].

The probabilistic machinery for the problem has now been defined apart from $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{t})$ in the posterior probability (28). In [7] two methods for estimating this probability are acknowledged. One is a variational approach, and the other is a computationally light approximation. The approximation assumes that the considered probability density can be presented well enough by multidimensional delta function centered at the most probable values for the precisions $\boldsymbol{\alpha}_{MP}$ and variance σ_{MP}^2 , that is, $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{t}) \approx \delta(\boldsymbol{\alpha}_{MP}, \sigma_{MP}^2)$. The same approach for estimation of probabilities is considered in [36], where the only difference mathematically is that $\boldsymbol{\alpha}$ has the same value for every weight. The procedure is based on the assumption that the properties of the distribution are rigid near the most probable values. There should not, for example, be sudden discontinuities in the distribution. In both sources the approximation is said to work well in practice [7, 36].

Now one can start using SBL for learning the precision of the noise β and the precisions of the priors for the weights $\boldsymbol{\alpha}$ by finding their most probable values. In SBL this is done by maximizing the marginal likelihood with respect to the precisions. Given the logarithms of $\boldsymbol{\alpha}$ and β , the expression to be maximized is defined in [7] as

$$\log p(\mathbf{t}|\log \boldsymbol{\alpha}, \log \beta) + \sum_{i=1}^N \log p(\log \alpha_i) + \log p(\log \beta). \quad (33)$$

By using the formula from [45] for the change of variables in a probability distribution, one obtains that $p(\log \alpha) = \alpha p(\alpha)$, which can be applied for the two latter terms in (33). As $p(\mathbf{t}|\log \boldsymbol{\alpha}, \log \beta) = p(\mathbf{t}|\boldsymbol{\alpha}, \beta)$ is specified by (29) and by removing terms not related to $\boldsymbol{\alpha}$ and β , the former expression defines the log-likelihood function in [7] as

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}, \beta) &= -\frac{1}{2} \left[\log |\beta^{-1} \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^T| + \mathbf{t}^T (\beta^{-1} \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^T)^{-1} \mathbf{t} \right] \\ &\quad + \sum_{i=1}^N (a \log \alpha_i - b \alpha_i) + c \log \beta - d \beta. \end{aligned} \quad (34)$$

This function will be maximized iteratively using the roots of the derivatives for $\boldsymbol{\alpha}$ and β . Correspondingly, the derivatives with respect to $\log \alpha_i$ and $\log \beta$ are given in [7] as

$$\frac{\partial \mathcal{L}}{\partial \log \alpha_i} = \frac{1}{2} \left[1 - \alpha_i (\mu_i^2 + \Sigma_{ii}) \right] + a - b\alpha_i, \quad (35)$$

$$\frac{\partial \mathcal{L}}{\partial \log \beta} = \frac{1}{2} \left[\frac{M}{\beta} - \|\mathbf{t} - \Phi \boldsymbol{\mu}\|^2 - \text{tr}(\Sigma \Phi^T \Phi) \right] + c - d\beta. \quad (36)$$

These derivatives are computed from an alternative presentation of the likelihood function presented later in 3.2.4 in equation (52), where the term $\boldsymbol{\mu}$ is assumed to be constant. The required derivatives for matrix expressions are available, for example, in [46]. Then solving for the critical points and noting that with a direct computation $\text{tr}(\Sigma \Phi^T \Phi) = \beta^{-1} \Sigma_i \gamma_i$, one obtains the update rules, which are the same as for the EM algorithm 1:

$$\alpha_i^{\text{new}} = \frac{1 + 2a}{\mu_i^2 + \Sigma_{ii} + 2b}, \quad (37)$$

$$(\sigma^2)^{\text{new}} = \frac{\|\mathbf{t} - \Phi \boldsymbol{\mu}\|^2 + (\sigma^2)^{\text{old}} \Sigma_i \gamma_i + 2d}{M + 2c}, \quad (38)$$

where $(\sigma^2)^{\text{new}}$ and $(\sigma^2)^{\text{old}}$ are $(\beta^{-1})^{\text{new}}$ and $(\beta^{-1})^{\text{old}}$ respectively [7]. These formulas are iterated until convergence or until maximum number of iterations to obtain the regularization for the weights in the model. The convergence is evaluated by monitoring the difference in the last and current estimates for $\boldsymbol{\mu}$.

After every iteration, the weights which have larger prior precisions $\boldsymbol{\alpha}$ than the threshold which was fixed before the iteration are pruned away by assigning the corresponding basis function with a coefficient zero as demonstrated in figure 8. This removes the corresponding basis from the iteration and usually makes the iteration more computationally efficient as well as numerically stable because there the precisions of the values would otherwise continue growing in the prior precision matrix \mathbf{A} making matrices involved in the computations more ill-conditioned. This pruning will, on the other hand, slightly restrict the convergence as the pruned values with large precisions cannot be modified to find a better optimum[7].

When the iteration is terminated, one can do predictions by computing the mean interpolation y_* and variance σ_*^2 for a prediction for at a new location \mathbf{x}_* . More specifically, the predictions can be computed by applying equation (30) for $p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2)$ and (24) for $p(\mathbf{t}|\mathbf{w}, \sigma^2)$ as well as the estimates $\boldsymbol{\alpha}_{MP}$ and σ_{MP}^2 in the integral expression

$$p(t_*|\mathbf{t}, \boldsymbol{\alpha}_{MP}, \sigma_{MP}^2) = \int_{\mathbb{R}^N} p(t_*|\mathbf{w}, \sigma_{MP}^2) p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}_{MP}, \sigma_{MP}^2) d\mathbf{w}, \quad (39)$$

where both probabilities $p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}_{MP}, \sigma_{MP}^2)$ and $p(t_*|\mathbf{w}, \sigma_{MP}^2)$ are Gaussian [7]. Hence, in [7] the resulting predictions are made with a probability distribution given as

$$p(t_*|\mathbf{t}, \boldsymbol{\alpha}_{MP}, \sigma_{MP}^2) = \mathcal{N}(t_*|y_*, \sigma_*^2). \quad (40)$$

Here the following values define the posterior mean and variance for the interpolation, given the learnt priors from iteration equations (37), as

$$y_* = \boldsymbol{\phi}(\mathbf{x}_*)^T \boldsymbol{\mu}, \quad (41)$$

$$\sigma_*^2 = \sigma_{MP}^2 + \boldsymbol{\phi}(\mathbf{x}_*)^T \Sigma \boldsymbol{\phi}(\mathbf{x}_*). \quad (42)$$

An advantage of SBL for interpolation is that the probabilistic approach provides a method of measuring uncertainty in the obtained mean interpolation through the associated variance [7].

3.2.2 Discrete cosine transform basis

As long as the basis $\phi(\mathbf{x})$ can be presented in a matrix form, it can be used for the model presented in (22). In practice, for dense matrices this becomes computationally too heavy for large problems as one would need to save a $M \times N$ matrix in memory, multiply it with its transpose, and then invert the resulting $M \times M$ matrix as in (31). In [7] it is stated that SBL is $\mathcal{O}(M^3)$ in complexity and $\mathcal{O}(M^2)$ in memory storage in general. For concreteness, if there are at least $M = 10^5$ basis functions and each measurement is given as a 32 bit number of 4 bytes, $M^2 \cdot 4B$ translates to 40 Gigabytes in memory which further implies that SBL would be impractical for the problem at hand without optimization of the algorithm itself [7].

The algorithm can be optimized for this problem by defining the basis using a transform function which will be inverted using an iterative method. As radar data is given in three-dimensions, a product of three discrete cosine functions is set as a basis. The discrete cosine functions are implemented using three-dimensional discrete cosine transform type-II algorithm (3-D DCT II) [47, 48]. More formally, given a signal in three dimensions $z(n_1, n_2, n_3)$ the 3-D DCT II with size $N_1 \times N_2 \times N_3$ outputs transformed signal $Z(k_1, k_2, k_3)$ which is defined as

$$\begin{aligned} Z(k_1, k_2, k_3) = & \sqrt{\frac{8}{N_1 N_2 N_3}} \varepsilon_{k_1} \varepsilon_{k_2} \varepsilon_{k_3} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \sum_{n_3=0}^{N_3-1} z(n_1, n_2, n_3) \\ & \cdot \cos\left(\frac{\pi}{2N_1}(2n_1 + 1)k_1\right) \\ & \cdot \cos\left(\frac{\pi}{2N_2}(2n_2 + 1)k_2\right) \\ & \cdot \cos\left(\frac{\pi}{2N_3}(2n_3 + 1)k_3\right), \\ & k_1 = 0, 1, \dots, N_1 - 1; k_2 = 0, 1, \dots, N_2 - 1; k_3 = 0, 1, \dots, N_3 - 1, \end{aligned} \quad (43)$$

where

$$\varepsilon_{k_i} = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } k_i = 0 \\ 1, & \text{otherwise} \end{cases} \quad i = 1, 2, 3, \quad (44)$$

[47].

Correspondingly, its inverse transform is defined as

$$\begin{aligned}
z(n_1, n_2, n_3) = & \sqrt{\frac{8}{N_1 N_2 N_3}} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \sum_{k_3=0}^{N_3-1} \varepsilon_{k_1} \varepsilon_{k_2} \varepsilon_{k_3} Z(k_1, k_2, k_3) \\
& \cdot \cos\left(\frac{\pi}{2N_1}(2n_1 + 1)k_1\right) \\
& \cdot \cos\left(\frac{\pi}{2N_2}(2n_2 + 1)k_2\right) \\
& \cdot \cos\left(\frac{\pi}{2N_3}(2n_3 + 1)k_3\right), \\
& n_1 = 0, 1, \dots, N_1 - 1; n_2 = 0, 1, \dots, N_2 - 1; n_3 = 0, 1, \dots, N_3 - 1
\end{aligned} \tag{45}$$

[47].

This formulation is beneficial as 3-D DCT II can be computed with three sequential 1-D DCT II algorithms each against one mode of the signal. Each of these are $\mathcal{O}(N_1 N_2 N_3 \log N_i)$, $i = 1, 2, 3$ in time complexity. Hence the matrix-vector multiplications via the 3-D DCT II with complexity $\mathcal{O}(N_1 N_2 N_3 \log N_1 N_2 N_3)$ is relatively efficient when compared to corresponding dense matrix-vector multiplication with complexity $\mathcal{O}((N_1 N_2 N_3)^2)$. Also, the DCT does not save explicitly the whole dense matrix Ψ to the memory storage, but rather it only needs $\mathcal{O}(N_1 N_2 N_3)$ of memory for the input values that are transformed. [47]

By vectorizing $Z(k_1, k_2, k_3)$, the 3-D DCT II transform matrix Ψ becomes orthonormal as it is created using the orthonormal 1-D DCT II algorithm. The orthonormality also defines the inverse transform matrix as Ψ^T . The 3-D DCT II transform matrix could be seen as a chained Kronecker product of three 1-D DCT II transforms as described in [49]. The definition of the transform as a matrix is possible as the efficient pruning of the bases during the iteration is carried out using a list of Boolean values. The Boolean values determine whether the basis will be pruned away without the need to handle memory heavy dense matrices [47, 48, 49, 50].

The 3-D DCT II transform matrix Ψ defines the basis as a square matrix where the size is defined by the number of basis vectors. As there are less measured locations than there are basis vectors, only the rows of Ψ corresponding to the measured locations are chosen. In compressed sensing literature this would mean that Ψ is multiplied from left by a sensing matrix $\mathbf{S} \in \mathbb{R}^{M \times N}$. The sensing matrix \mathbf{S} maps N locations to be interpolated to the corresponding measured locations M . To accomplish this, if there are several values at the same measured location, a mean of the values is taken and then used for measurement at the given location [51].

In practice, the matrix \mathbf{S} is computed efficiently using consecutive stable sorting algorithm to determine values on a common grid cell and by then taking the mean of those grid cells. The resulting matrix is denoted by $\Phi = \mathbf{S}\Psi \in \mathbb{R}^{M \times N}$, where $N = N_1 N_2 N_3$. For computational efficiency multiplication by the matrix \mathbf{S} is implemented again by another list of Boolean values, and multiplication by its transpose corresponds to zero padding into a larger N -dimensional vector, which can be done using the Boolean list as well.

The former approach is more efficient than the approach in [32] where it is instructed into compute the basis explicitly as a dense matrix to memory before running a similar compressive sensing algorithm, which in top of the memory issues would also lead to $\mathcal{O}(M^3)$ matrix inversion. With a method for estimation of the diagonal of an inverse matrix and CG, the algorithm can be run more efficiently computationally and memory-wise [11]. This approach could be used to optimize the variational algorithm in [32] as well as in [7].

3.2.3 Matrix inversion and diagonal of the inverse

CG can be used to compute solutions to linear systems with symmetric positive-definite matrices. The matrix $\Sigma^{-1} = \sigma^{-2}\Phi^T\Phi + \mathbf{A}$ in (31) is a symmetric matrix, and also a positive definite matrix due to the fact that \mathbf{A} is a diagonal matrix with positive elements, and Σ^{-1} is a sum of two positive non-zero dot products:

$$\mathbf{x}^T \Sigma^{-1} \mathbf{x} = \sigma^{-2} \mathbf{x}^T \Phi^T \Phi \mathbf{x} + \mathbf{x}^T \mathbf{A}^{1/2} \mathbf{A}^{1/2} \mathbf{x} > 0, \quad \text{for every } \mathbf{x} \neq 0. \quad (46)$$

Hence, the usage of CG for memory-friendly matrix-vector multiplication by Σ is possible. CG for a symmetric positive-definite matrix Σ^{-1} is essentially gradient descent with the descent directions being Σ^{-1} -orthogonal, or more formally, with two different search directions $\mathbf{d}_i, \mathbf{d}_j$ it holds that $\mathbf{d}_i^T \Sigma^{-1} \mathbf{d}_j = 0$. [11].

As matrix-vector multiplications are the most expensive operations in CG, with time complexity of order $\mathcal{O}(N \log N)$ in the given setting, and CG converges asymptotically with a multiplier of the square root of the condition number of Σ^{-1} , that is $\sqrt{\kappa}$, this approach is beneficial computation-wise as gradient descent converges asymptotically with a corresponding multiplier of κ . The overall convergence of CG for computation of the matrix-vector product $\Sigma \mathbf{b}$, where $\mathbf{b} \in \mathbb{R}^N$, is $\mathcal{O}(\sqrt{\kappa} N \log N)$ in time complexity and the memory usage of the algorithm is asymptotically $\mathcal{O}(N)$ [11].

To use CG in a space where the number of dimensions corresponds to the number of data locations M , matrices are inverted using the Woodbury matrix identity

$$\Sigma = (\sigma^{-2}\Phi^T\Phi + \mathbf{A})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\Psi^T \mathbf{S}^T (\beta^{-1}\mathbf{I} + \mathbf{S}\Psi\mathbf{A}^{-1}\Psi^T \mathbf{S}^T)^{-1} \mathbf{S}\Psi\mathbf{A}^{-1}, \quad (47)$$

where the complexity stays the same with CG due to matrix inversion with DCT as a subroutine. This is done as preliminary tests seemed to lead to faster convergence of CG when inverting through the identity (47). [46].

Now matrix-vector multiplications by Σ can always be computed using CG. This is efficient enough for computation of coefficients in the iteration in equation (32) in a reasonable time, but in equations (35) and (36) the computation of diagonal elements of Σ would be too computationally expensive, if the diagonal values were computed by explicitly computing the whole matrix. Hence the diagonal is estimated using operator probing [8, 10].

The estimate of the diagonal of Σ , that is \mathbf{D}^s , given vectors $\mathbf{v}_k, k = 1, \dots, s$, with

certain properties discussed later, is given as

$$\mathbf{D}^s = \left[\sum_{k=1}^s \mathbf{v}_k \odot \Sigma \mathbf{v}_k \right] \oslash \left[\sum_{k=1}^s \mathbf{v}_k \odot \mathbf{v}_k \right], \quad (48)$$

where \odot and \oslash are element-wise multiplication and division respectively [10]. It is shown in [10] that if the vectors $\mathbf{v}_k, k = 1, \dots, s$, are chosen as column vectors from a matrix with its row vector at index i being orthogonal to all rows j , where the matrix to be estimated has nonzero values, the estimate will give exact result for the diagonal. If the number of chosen vectors s is set to be significantly lower than the largest dimension of the matrix to be estimated, columns \mathbf{v}_k should be chosen from a matrix satisfying the aforementioned property as well as possible. As certain number of columns from a Hadamard matrix satisfy this property quite well, as can be seen from the visualization in the figure 9, the vectors \mathbf{v}_k are extracted from the columns of the Hadamard matrix [10].

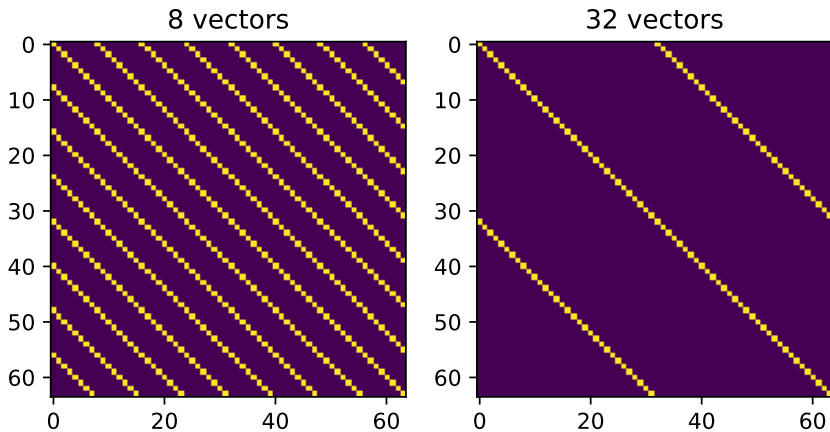


Figure 9: The locations of the values that are extracted from a 64-by-64 matrix by the diagonal estimator in operator probing using different number of Hadamard vectors. Given a row the value of the diagonal will be computed by summing the yellow values. The values computed will not be exact if the values at off-diagonal are nonzero, but the approximation might still be enough to provide sensible information about the variance [8, 10].

As the basis vectors are chosen originally as orthonormal, the off-diagonal values of Σ could possibly tend to be smaller than the diagonal values. This is due

to orthonormal vectors having larger variance than covariance with a mutually orthonormal vector when compared with respect to the absolute value. Hence the effect of the off-diagonal values caused by the column vectors from the Hadamard matrix might have a lesser impact as demonstrated in [10]. If it happens that the off-diagonal values do not decay, the convergence of $\boldsymbol{\alpha}$ and σ^2 might be harder as the approximations obtain noise from correlations between the basis vectors. The effect of these correlations could be countered with a larger number of vectors \mathbf{v}_k as fewer nonzero values affect the approximation. Even though at some iterations the approximation is bad, it is still possible that the method will eventually find a good approximation because the values change as the iteration progresses, given that the bases affected by the correlations are not pruned away [10].

3.2.4 Computation of the cost function

According to [35] every iteration of a working EM algorithm should only increase the value of the log-likelihood. Hence, to evaluate the convergence and correctness of the algorithm, the log-loss function (34) should be computed if its computation is inexpensive enough. To ease the computation of the log-loss function, the terms containing matrix-vector products are modified for simpler computation. The following four formulas are all from [7]. First by applying the determinant identity

$$|\mathbf{A}||\beta^{-1}\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T| = |\beta^{-1}\mathbf{I}||\mathbf{A} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi}|. \quad (49)$$

The application of determinant identity (49) modifies the logarithm as

$$\log |\beta^{-1}\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T| = -\log |\boldsymbol{\Sigma}| - M \log \beta - \log |\mathbf{A}|. \quad (50)$$

Then Woodbury matrix inversion identity is applied to the other term, and the obtained expression is further modified to obtain

$$\mathbf{t}^T(\beta^{-1}\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T)^{-1}\mathbf{t} = \beta\mathbf{t}^T(\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu}) = \beta\|\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 + \boldsymbol{\mu}^T\mathbf{A}\boldsymbol{\mu}. \quad (51)$$

Combining these results will lead to an overall cost function

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}, \beta) = & -\frac{1}{2} \left[\beta\|\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 + \boldsymbol{\mu}^T\mathbf{A}\boldsymbol{\mu} - \log |\boldsymbol{\Sigma}| - M \log \beta - \log |\mathbf{A}| \right] + \\ & + \sum_{i=1}^N (a \log \alpha_i - b\alpha_i) + c \log \beta - b\beta. \end{aligned} \quad (52)$$

Now only the computation of $\log |\boldsymbol{\Sigma}|$ is complicated as all terms are relatively simple to compute as modifications of previous terms. The log-determinant of $\boldsymbol{\Sigma}$ is evaluated using an approximation to produce a relatively inexpensive estimate of the cost function. The approximation is defined by the integral

$$\Delta = \int_0^1 \text{tr}[\mathbf{N}(\mathbf{D} + t\mathbf{N})^{-1}]dt + \ln \det(\mathbf{D}), \quad (53)$$

where \mathbf{D} and \mathbf{N} are the diagonal and off diagonal of $\boldsymbol{\Sigma}$ respectively, and t is just a integration variable [12].

The trace and the diagonal in equation (53) can be evaluated efficiently using operator probing and CG. The integral is computed using Simpson’s rule after the trace is computed for an evenly divided set of points on the interval $[0, 1]$. The quality of the approximation depends on the number of points on the interval as well as on how many vectors are used for operator probing. As the cost is only computed to give an idea of convergence and whether the iteration increases the marginal posterior likelihood, the errors do not affect the iteration and hence do not matter as much as earlier with the operator probing of the diagonal of Σ [10, 11, 12].

To conclude the discussion on the details of SBL, the whole algorithm is presented in 2 for clarity [7, 10, 11].

Algorithm 2 SBL algorithm

Input:

- Measurements $\{\mathbf{x}_n, t_n\}_{n=1}^M$, where M is the number of measurements and \mathbf{x}_n, t_n are the locations and values of the measurements, from weather radar with the bins anomalous propagation and biological scatterers filtered out.
- Threshold alpha, initial values of the precisions α_{MP} and β_{MP} , number of Hadamard vectors, shape and resolution of the raster, maximum number of CG iterations, CG threshold, maximum number of EM iterations.

Output:

- Approximate mean interpolation raster and standard deviation raster \mathbf{y}, σ respectively.
- 1: Compute the mean of the data points at each raster cell and a Boolean array to represent non-empty locations in form of a sensing matrix \mathbf{S} described in section 3.2.2.
 - 2: Set starting values for α_{MP} and β_{MP} .
 - 3: iterate the EM algorithm with operator probing for posterior variances and zero prior pruning until convergence of $\boldsymbol{\mu}$ from (32) or until maximum iteration count of the EM algorithm using the formulas (37).
 - 4: Compute the mean interpolation estimate \mathbf{y} (41) using the learnt posterior mean $\boldsymbol{\mu}$.
 - 5: Compute the standard deviation estimate σ (42) with operator probing (48).

Return: \mathbf{y}, σ

3.3 Experimental setup

SBL is tested in three experiments using the attenuation corrected and echo filtered volume from the given timestamp described in section 3.1. In the first approach, interpolation is performed onto a large 256x256x64 raster in Cartesian coordinates. In the second approach interpolation is performed onto a CAPPI 256x256 raster created from data between heights of 2000 m and 2100 m, where the altitudes for the radar bins are specified by the formulas (10) and (11). Then in the last approach, only a vertical slice from azimuth angles at 0 and 180 degrees is used for interpolations

onto an RHI 256x64 raster. These rasters correspond to vertical resolution of 650 m and horizontal resolution of 100 m. RHI and CAPPI are visualised in the whole three-dimensional space in figure 10.

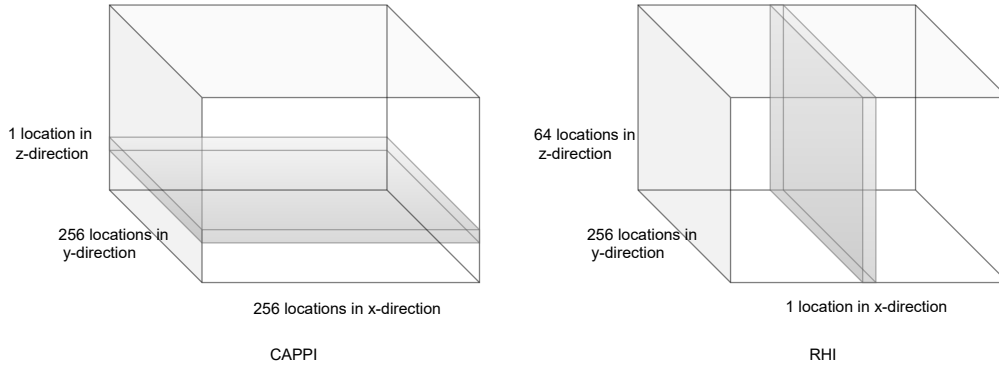


Figure 10: Here CAPPI and RHI onto which interpolation is performed are visualised in the whole three-dimensional space. The gray areas are the corresponding interpolated two-dimensional rasters in the second and third interpolation cases in the experiments, while in the first approach interpolation is made onto a three-dimensional raster and only visualised on similar gray area slices at given altitudes.

Training and test sets split by elevation angles are presented for the experiments as in blue and red curves, respectively, in figure 11 for the three-dimensional raster as well as for CAPPI and RHI rasters. Reflectivity values of 0 dBZ are set to the radar bins at positions in which there are no radar returns. This is done as the method has to present all the values numerically in each of the raster locations. The interpolation is done in logarithmic units of reflectivity, and the mean is taken between values that would be at the same location for the matrix \mathbf{S} in training, and for every raster location in the test set. The mean of the measured values for belonging to each raster cell has to be taken as otherwise the values could not be presented in the locations specified by the basis vectors, and hence the model could not be trained or evaluated. For the training set of the large three-dimensional raster, a mean is taken between the first sweep and the third sweep of the lowest elevation angle, but otherwise only the first example of the elevation angle in the volume is used. For CAPPI and RHI only the first of the sweeps is used for training. In all cases, the test set uses only the first sweep with the corresponding elevation angle. The lowest and highest angles are included in the training set to make sure that the models interpolate that lie in the test set.

The performance of SBL is tested against OK, NN, IDW and linear interpolation. The corresponding interpolations are created using the basic interpolation techniques available in wradlib. Due to computational cost of matrix inversion, OK is done in pieces defined by the 12 nearest neighbors at a given point, and hence the standard deviation raster is not created. Also, due to computational burden, the three-dimensional raster is not computed using linear interpolation from wradlib. The different methods are compared using RMSE as a metric, by comparing the measured

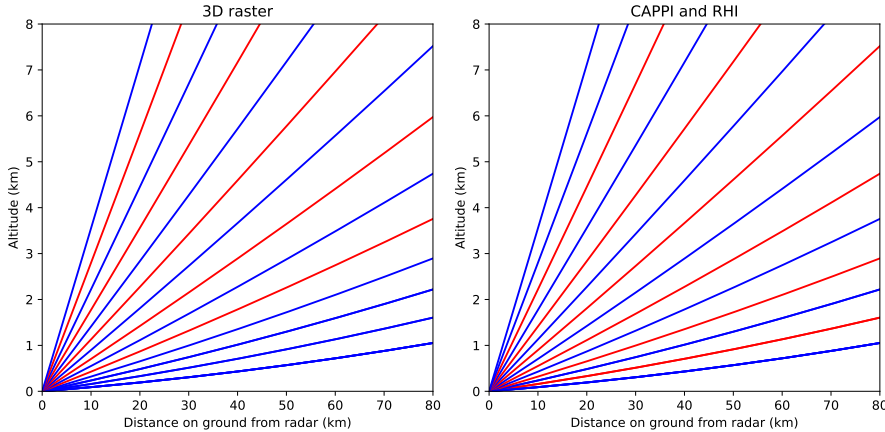


Figure 11: A visualization of dataset splits for the three-dimensional raster on the left and for CAPPI and RHI on the right, and the propagation of rays based on equations 10 and 11. The elevation angles are from lowest to highest 0.5, 0.9, 1.3, 2.4, 3.1, 4.0, 5.1, 6.4, 8.0, 12.5, 15.6, and 19.5 degrees as in VCP 212 scanning strategy. The angles given in blue are used in the training of the interpolation methods, while the red ones are used for evaluation of the interpolation errors in interpolations onto the rasters.

values from the test set at the locations of the raster, as well as qualitatively assessing the resulting interpolations [9].

The large SBL raster is computed using 10000 as the precision threshold for pruning and with 64 Hadamard vectors used for operator probing. CAPPI uses precision threshold of 10 and 256 Hadamard vectors. RHI has the same precision threshold as the large raster, but for pruning there were 1024 Hadamard vectors. For RHI, log-likelihood is computed every tenth iteration to provide details on the convergence. The approximation (53) uses ten points for the Simpsons rule and 1024 Hadamard vectors for estimation.

The iteration for the large raster is terminated after maximum number of 300 iterations of the EM algorithm, i.e., algorithm 1, while for CAPPI raster the algorithm is terminated after 100 iterations and RHI after 50 iterations. Each run has tolerance of 10^{-3} for the convergence of the EM algorithm and the convergence tolerance for CG is 10^{-8} times the size of the vector which is multiplied by the inverted matrix to find the solution. Starting values for α and σ^2 are a full vector of ones, and the estimated variance of the dataset, respectively.

4 Results

The results from the experiments are presented in this section, first the results for interpolation onto a three-dimensional raster, then onto a CAPPI raster, and finally onto an RHI raster. The iterations did not converge to the given tolerance after the maximum number of iteration in any of the cases.

4.1 Interpolation onto a three-dimensional raster

A slice of the three-dimensional mean interpolation raster from an altitude interval between 2000 and 2100 meters returned by SBL is presented on the left in figure 12 followed by the input data for SBL in the middle and the approximate standard deviation raster on right from the same altitude interval. This computation was made to a raster of size $256 \times 256 \times 64$, and therefore the values below and above the interval in the whole raster are also taken into account during interpolation. Furthermore, the values in the input data are computed through the mean of the values that belong to the given grid cell as specified in the section 3.3 for the test and training data.

The mean interpolation on the left seems to fit to the higher reflectivity values of the input data, and hence the continuous shape of the input data on the upper right corner can be recognized from the mean interpolation raster. SBL seems to succeed to model the uncertainty of the interpolation as on the standard deviation raster, the locations further from input data have higher standard deviation when compared to locations near locations of input data.

High reflectivity values seem to be located on fainter areas of standard deviation plot in general. This could be an effect of the echo class filtering, where the highest reflectivities correspond to meteorological events, which were not filtered out. Another reason could be that the values are similar locally leading to a more robust mean interpolation raster.

There is a continuous area of standard deviation with values over 150 dBZ on the bottom left side of the raster. The same area seems to have larger reflectivity values than the empty area with continuous high standard deviation on the bottom right section of the raster. These areas have fewer values in the input data due to echo class filtering and hence the standard deviations are higher.

The interpolated values seem to be more sharp overall for SBL than for NN, IDW and OK as given in figures 12 and 13, where the rasters from the same altitude are presented. The areas of high reflectivity seem to be quite similar for all of the images, and similar areas are recognized especially in the upper section of the raster.

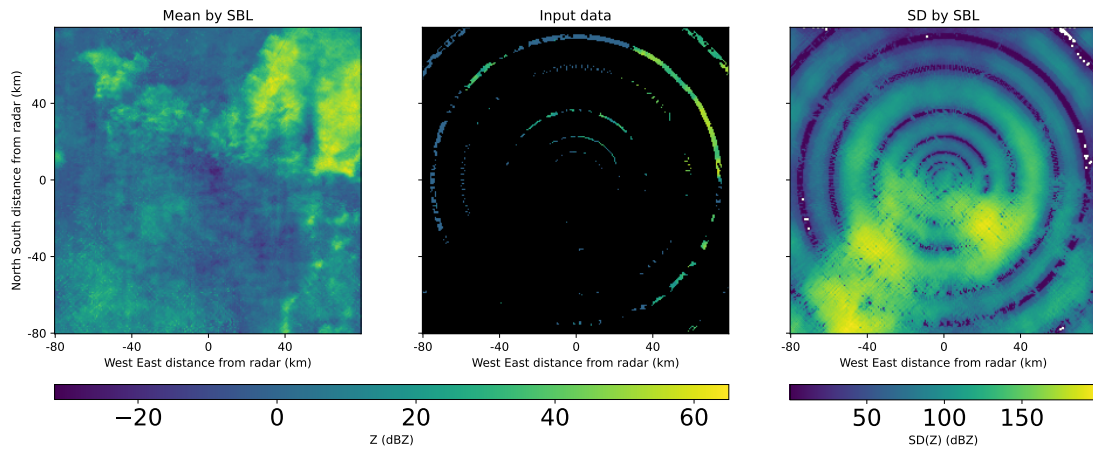


Figure 12: On the left a CAPPI raster from altitude of 2 km produced by SBL is presented. In the middle the input data mapped to correct positions using \mathbf{S}^T is visualized at the same altitude, and on the right is the approximate standard deviation produced by SBL. In the middle black color represents points that correspond to the locations from which there are no training or input data as specified by the split in section 3.3, while on the right white values represent errors due to negative variances produced by SBL. The negative values occur due to the influence of the off-diagonal values during operator probing, and the square root operation transforms these values to represent missing data values. The timestamp of the volume used for interpolation is 9th of June 2019 at 19.03:20 UTC.

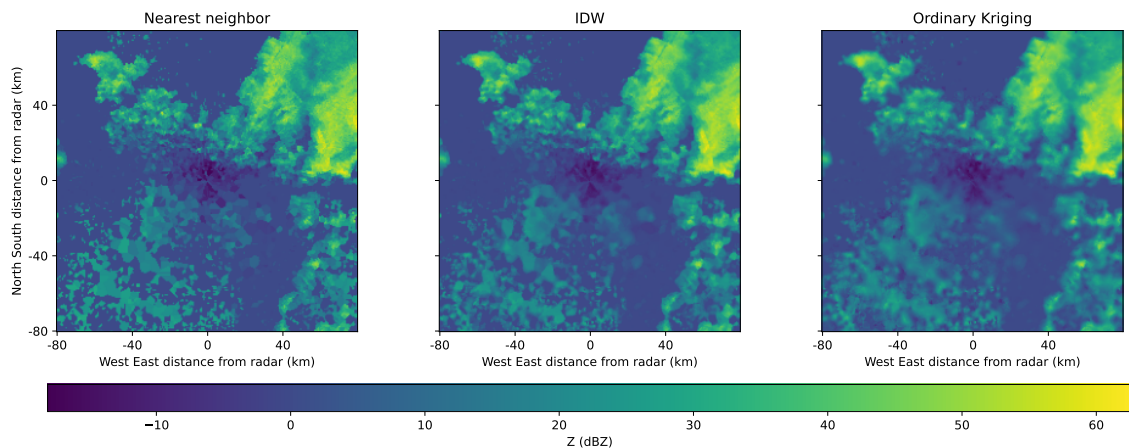


Figure 13: Here the NN, IDW, and OK interpolations from altitude of 2000 m are presented from left to right. The solution in OK is clearly the smoothest followed by IDW. As all methods are based on interpolation using nearest neighbors the solutions seem quite similar. The timestamp of the volume used for interpolation is 9th of June 2019 at 19.03:20 UTC.

The gaps between the areas of high reflectivity, as in the the bottom left section of the raster, are not present in the interpolation by SBL, but the area has quite high standard deviation according to SBL, possibly due to relatively bad convergence. The areas with low reflectivities are more explicitly assigned with similar low values by the nearest neighbor based techniques in other methods, as can be seen in the noise free area between the high reflectivities on the upper area of the raster in figure 13 when compared to figure 12. The noisiness of SBL at values without radar echoes is due to inexact interpolation, unlike the approaches using exact nearest neighbors approaches as subroutines. Also, as the values for locations without an echo return were set to zero dBZ and interpolation is done on a logarithmic scale, the values on regions with no or very faint echoes, possibly have fluctuations due to some echoes being of smaller or larger nearby. These noisy fluctuation could have been removed with a significantly lower dBZ value for the locations without an echo return, as the small values would represent the lack of scatterers locally with respect to other values in the obtained raster. Hence, the corresponding locations seem to have some slight fluctuations in the reflectivity values.

In figure 14, RMSE for each CAPPI produced on consecutive non-overlapping altitude intervals of hundred meters in height are compared between NN, IDW, OK and SBL. It can be seen that with this data OK gives the best overall performance on most intervals, and NN gives the worst performance all the way up to 4700 meters.

Between 0 and 2800 meters as well as between 4200 and 4500 meters, IDW and SBL perform relatively similarly with each beating the other on different altitude intervals. IDW in total has the second best performance in the majority of the height intervals as it clearly outperforms all the methods except OK between 2800 and 4200 meters as well as upwards from 4600 meters. The magnitude of RMSE seems to be twice as large at 0 and above 4500 meters compared to intervals in between. This is most likely due to the fact that there are less data at those heights than the others. Also, another phenomenon that affects the results is that as the interpolation moves higher, the lower elevation angles move further in horizontal distance than the 80 km distance threshold and the data thus becomes sparser in density as the altitude increases as can be seen in figure 11.

Also, the clear increase in RMSE in figure 14 at about 4500 meters seems to happen at the bottom of the altitude of the melting layer. It could be that the values below the melting layer are affected by values that are in the melting layer in the resulting interpolations. Possibly measured empty echoes are compared with interpolated high reflectivity values from the melting layer leading to higher difference for RMSE.

The quality of the interpolation given by SBL at the higher altitudes is quite bad. For example, at height interval of 6200 to 6300 meters in figure 15, the method seems to be influenced by the input values from lower altitudes. This is possibly due to the fact that there are only a few input values above the interpolated locations resulting to the nearest value unoptimally extrapolating the location. Another possible reason could be that the EM algorithm has not converged well enough at the 300th iteration, and hence the results are as above.

The interpolation on a RHI raster at azimuths between 0 and 180 degrees is

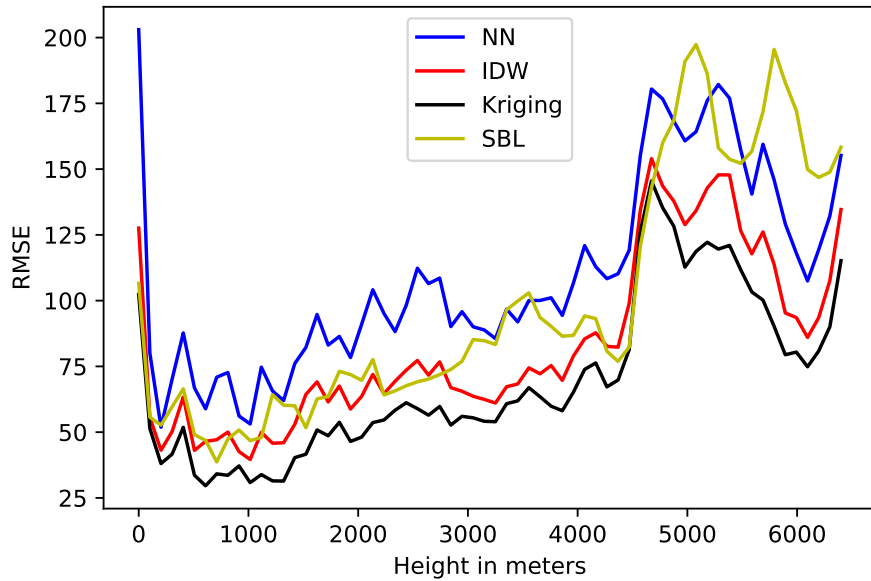


Figure 14: RMSE of the interpolation on the large raster as a function of height measured from the antenna of the radar. The interpolation is computed from a volume with timestamp of 9th of June 2019 at 19.03:20 UTC. Here the methods compared are NN, IDW, OK, and SBL. SBL out-performs IDW at certain heights but as the height increases above 4500 meters, all the methods start to work significantly less precisely and SBL no longer competes with IDW. IDW seems to outperform SBL most of the time, while NN seems often to be the worst. Overall OK performs the best with the smallest RMSE everywhere.

presented in figure 16. Kriging performed the best with RMSE of 67.68 followed by SBL with a slightly lower value 67.82, then by IDW with 80.33, and by NN with 93.42. The results seem to be quite similar to the CAPPI interpolation results in figure 14, but this time the performance of IDW is significantly worse than for SBL. However, SBL took impractically long to iterate the 300 iterations with a run time of 4 days and 12.5 hours. This makes SBL still quite impractical as OK took 5 minutes and 36 seconds to interpolate the data, and NN and IDW took 12.7 and 3.2 seconds respectively. The results for NN and IDW are strange as IDW uses NN as a subroutine, but the run time for NN was more than for IDW, hence this result is possibly due to some software optimization problems in wradlib.

The input data seem to correspond well to the interpolated rasters. SBL has a continuous area of high reflectivity with some stripe like artifacts everywhere, while the other methods have a continuous area too with artifacts from nearest neighbors based approaches. The melting layer is clearly visible for each of the methods as well as an area of moderate precipitation with reflectivity of about 35 dBZ on the right side section of the rasters.

The standard deviation raster has its highest values below the lowest elevation angle which is understandable because the area is extrapolated. The other areas

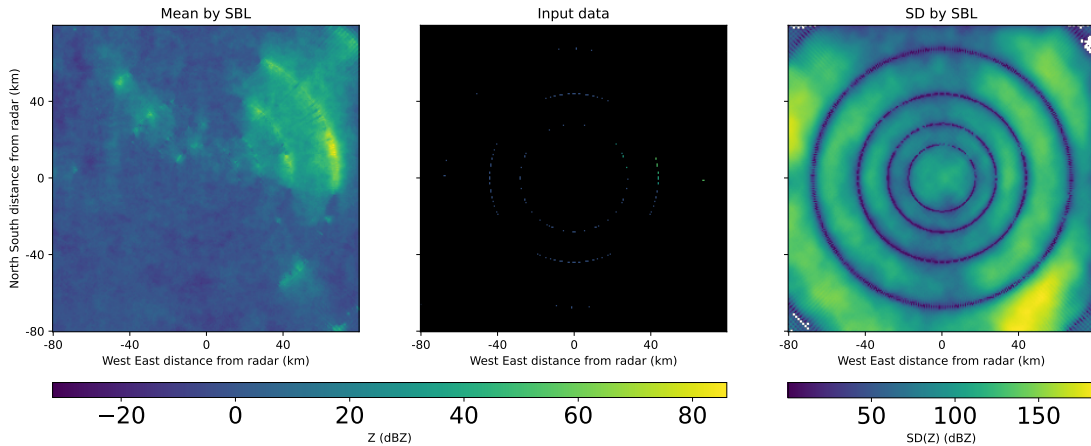


Figure 15: On the left, a CAPPI raster from altitude of 6.2 km produced by SBL is presented. In the middle the input data mapped to correct positions using \mathbf{S}^T is visualized at the same altitude, and on the right the approximate standard deviation produced by SBL is illustrated. In the middle, black color represents the locations to which the method interpolates. The timestamp of the volume used for interpolation is 9th of June 2019 at 19.03:20 UTC.

of higher standard deviations, though in smaller magnitude than below the lowest elevation, are again possibly due to gaps from echo classification, the form of the local data, and unoptimally converged EM iteration.

4.2 Interpolation onto a CAPPI raster

Interpolation results for CAPPI are presented in figure 17, where the first row shows only the approximate standard deviation raster produced by SBL. The second row begins with the mean interpolation raster from SBL on the left, and continues with NN and IDW. The last row from left to right shows linear interpolation, OK, and the data that is used as input to SBL.

Table 1: RMSE scores for two dimensional CAPPI and RHI rasters with methods SBL, OK, linear interpolation, IDW, and NN. OK works best in each class

RMSE	SBL	OK	Linear	IDW	NN
CAPPI	337.1	124.9	150.1	158.5	178.1
RHI	83.80	56.80	59.47	62.01	86.57

RMSE scores are presented in table 4.2. OK performs the best again with RMSE of 124.9 followed by linear interpolation with RMSE of 150.1. The next best performers are IDW and NN with scores of 158.5 and 178.1, respectively. SBL performs the worst with RMSE of 337.1. The results seem to have a general trend

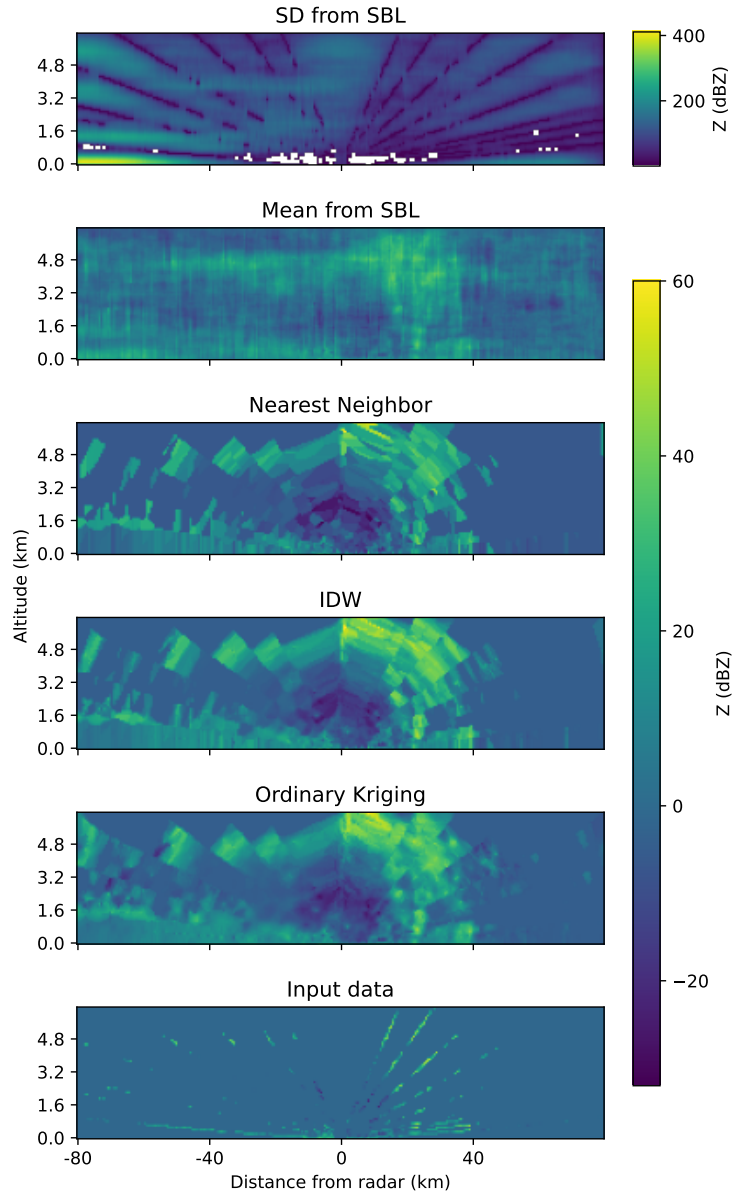


Figure 16: RHI slices created from the interpolated large raster. From top the first and the second plots are the standard deviation and the mean interpolation outputs from SBL, followed by NN as well as IDW, OK and finally a plot of the input data mapped to correct positions using \mathbf{S}^T . The timestamp of the volume used for interpolation is 9th of June 2019 at 19.03:20 UTC.

that the smoothest solution performs the best, similarly as for interpolation onto the three-dimensional raster. The difference compared to the three-dimensional raster is that SBL gives significantly worse results than the others, which is explained by SBL not having converged yet at the given 100 iterations. The iteration of SBL took about 5 hours and 45 minutes, while OK took about 4.3 seconds and the other

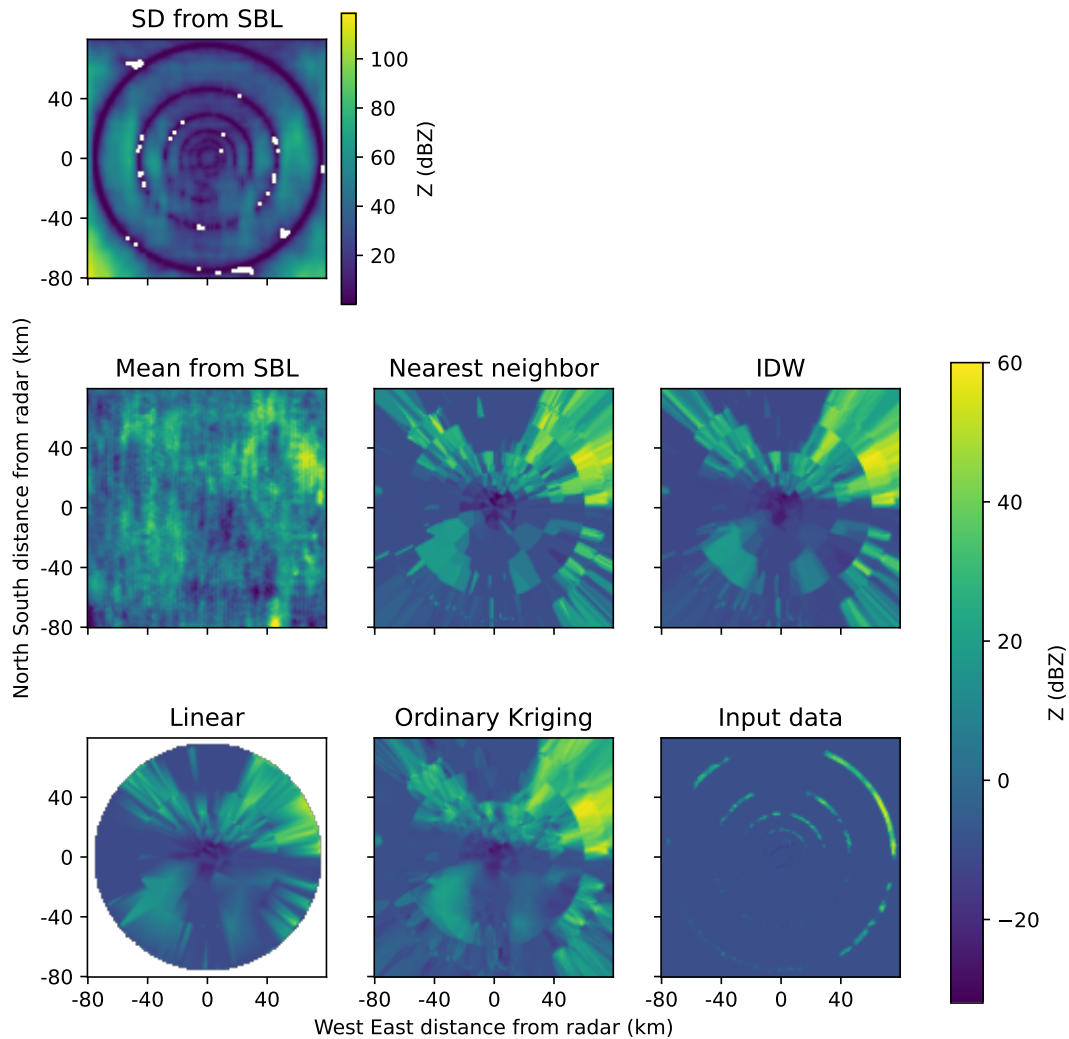


Figure 17: In this figure CAPPIs produced using input data from altitudes between 2000 and 2100 meters are presented. In the first row there is the standard deviation raster for the CAPPI output by SBL. Second row starts with the mean interpolation raster output by SBL, followed by the interpolated rasters by NN and by IDW correspondingly. On the last row, the linear interpolation, followed by OK, and input data to SBL mapped to correct positions using \mathbf{S}^T . The SBL rasters were computed with 100 iterations, pruning threshold of 10, and with 256 probing vectors. The timestamp of the volume used for interpolation is 9th of June 2019 at 19.03:20 UTC.

methods took about 114 ms, 108 ms and 52 ms for linear, NN and IDW, respectively.

All interpolations seem to have similar kinds of high reflectivity areas on the upper side and left side of the raster as for the large three-dimensional raster earlier. The interpolations based on the nearest neighbors approaches have now more clear artifacts resulting from the scans conducted in the radial and azimuth dimensions.

The large gaps between the values that form from the missing elevation angles affect the quality significantly as there are no data from above or below.

The two-dimensionality of the data might lead to strong effects on the quality of the interpolation produced by SBL, but a more likely reason for the worse results is the failed convergence of the method as the results are based on only 100 iterations and with prior precision threshold of 10, which affects the convergence as some values are pruned early. In general, smaller threshold should mean smoother result as coefficients with small weights are pruned, but as the method does not converge, the effect of this parameter is difficult to analyze.

The standard deviation seems to correspond to the locations of the input data. The large standard deviation values on the bottom left side of the image are not present in the same magnitude, which probably is due to the interpolation being agnostic of the values below and above the 2000-2100 meter interval. Still the values at the corners are higher as the raster is extrapolated to those regions.

4.3 Interpolation onto an RHI raster

For RHI raster the interpolation results are illustrated in figure 18, and the corresponding RMSE scores are presented in table 4.2. Once again, OK performed the best with RMSE of 56.80, followed by linear interpolation and IDW with RMSEs of 59.47 and 62.01, respectively. SBL got RMSE of 83.80, beating only nearest neighbor with RMSE of 86.57. The time that it took to iterate the 50 iterations of EM algorithm with the log-likelihood estimates every tenth iteration and with 1024 probing vectors was about 29 minutes for SBL, while all the other methods except OK, interpolated in less than 50 milliseconds. OK took about 1.6 seconds.

The interpolated raster created by SBL does not fit too close to the maximum values of the input data as the iteration does not converge in the given maximum number of iterations. The melting layer is a lot blurrier in the raster produced by SBL, but it is still faintly noticeable near 4800 meters. The same conclusion about the height of the melting layer holds for all the other methods as well. The low number of iterations and the fact that there are significantly fewer data points could be reasons why the results for SBL are now worse. Some of the values that probably correspond to moderate rain with reflectivity of about 30 dBZ are visible on the right side of the raster. These values are not visible as strongly for SBL as for the other interpolation methods or even the input data itself. Overall, the bad convergence causes the interpolant to have quite a lot of noisy dBZ values corresponding to at least light rain all over the raster.

The standard deviation raster produced by SBL has a lot more erroneous negative values than in the other cases presented above. In general, it seems that there are more problems with the approximation in the smaller rasters than in the larger one. This could be due to the basis vectors becoming smaller in dimension, which results in covariances being larger as there are fewer points used in the iteration where the values in basis vectors differ.

Finally, the computation of the approximate cost function is presented in figure 19. It can be seen that the log-likelihood (52) seems to increase for the 20 first

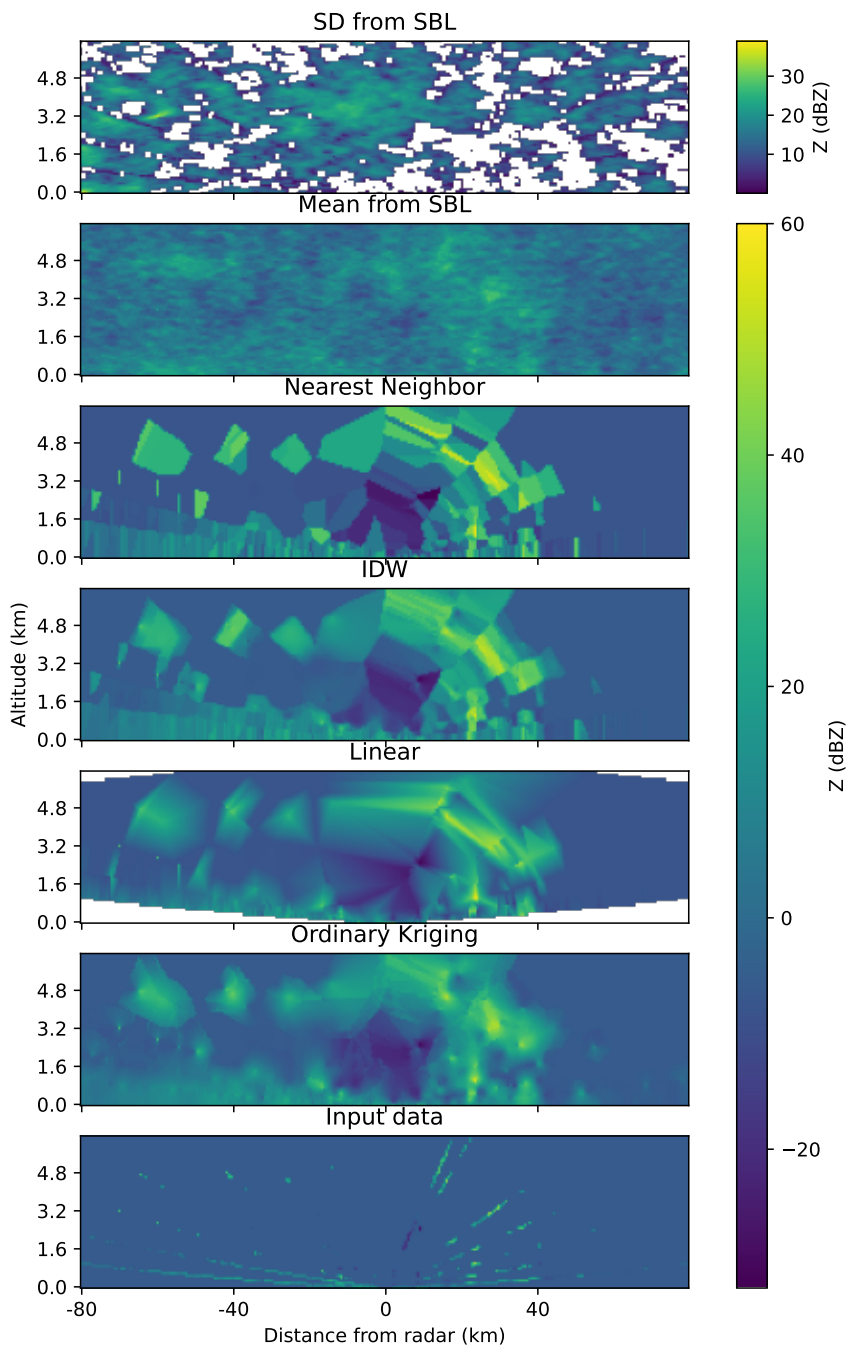


Figure 18: RHI plots from the different interpolation methods. First is the standard deviation and mean interpolation after 50 iterations of SBL with threshold 10000 for pruning and 1024 vectors for operator probing. The third raster is the result of NN followed by IDW, then there is the result of linear interpolation and as second to last of OK. The final picture is the input data used for interpolation mapped to correct positions using \mathcal{S}^T . The only method taking all the data into account in each point is SBL as all the others collect information locally only from a set of nearby points. The timestamp of the volume used for interpolation is 9th of June 2019 at 19.03:20 UTC.

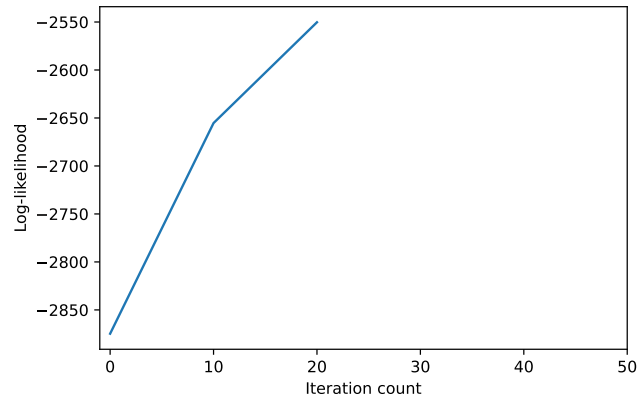


Figure 19: A plot of the increase of log-likelihood for the interpolation onto an RHI raster. Only three points from the iteration are visible once the rest of the iterations contain negative prior precision values, resulting to failure in the approximation of the likelihood.

iterations. Then at 30 iterations the approximation starts to have negative prior precision values, which makes further computations fail. It could be stated that the employed method for estimating the logarithm of determinant (53) is not too useful for SBL when the iteration is not stable. At the end of the 50 iterations of the method, there are over 106 negative prior precision values with the worst value being about -6.9×10^5 . For comparison, the large raster after 300 iteration had no negative values for the prior precisions and neither did the interpolation onto the CAPPI raster after 100 iterations. As the computation of the log-likelihood is problematic, the model comparison provided by the Bayesian framework on data analysis can be used only implicitly through the learning of locally optimal precisions.

5 Discussion

During the iteration of SBL, the matrix inversion formula (31), computed using the Woodbury matrix identity (47), acts as the most heavy operation complexity-wise. The reason for this is that as the components of the prior precision α are learnt during iteration, the precision matrix might become ill-conditioned. The sorted values of the priors in interpolation onto the three-dimensional raster are presented in figure 20, and as can be seen, the smallest coefficients are of order 10^{-8} (1/dBZ) and the largest values are set by the prior precision threshold to 10^4 (1/dBZ). This difference might lead to a matrix whose eigenvectors are scaled differently with smaller and larger eigenvalues, and as the convergence of CG is dependant of the condition number of the matrix, computed as the fraction of the largest and the smallest eigenvalues, the iteration becomes slow.

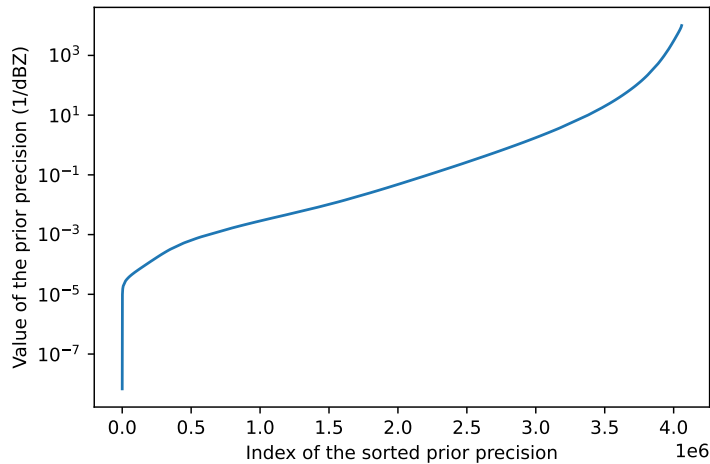


Figure 20: The sorted magnitudes alpha precision parameters on logarithmic scale after the end of the iteration for the large $256 \times 256 \times 64$ raster with precision threshold 10000 (1/dBZ). As the larger values correspond to larger likelihood of the coefficient being zero, and the values passing threshold corresponds to certain zero, most of the basis functions have a high probability of having a large coefficient. At the end of the iteration about 100000 of the coefficients have been explicitly pruned away.

Another problem during the iteration is that the posterior covariance matrix Σ from (31) might lose its positive-definiteness: the operator pruning may lead to erroneous negative values, which in turn leads to negative prior precisions. The negative values potentially also slow the CG, possibly preventing it from converging correctly. Another effect of these negative posterior covariance values is also visible in figure 12, where their effect leads to missing values for standard deviations.

It was also noticed during setting up the iteration that for smaller rasters there were artifacts at certain rows in the obtained raster. These artifacts did not seem to disappear with more iterations, but adding more pruning vectors helped. Hence, it

might be that for smaller rasters the covariances between different basis vectors are larger as there are fewer locations where the basis vectors could have differing values.

Hence, more Hadamard vectors are possibly needed for better approximation on smaller rasters than for the large raster as the more there are Hadamard vectors, the less significant is the effect from off-diagonal values of the covariance matrix.

Because reflectivity is modeled in SBL as a normal random variable as given in (23), using the errors from the interpolation, the validity of this normality assumption can be evaluated. The standard scores, or Z-scores, for the errors \hat{e} , i.e., the differences between the mean interpolations and the mean test data values at a specific location, are computed as

$$Z = \frac{\hat{e} - \hat{\mu}}{\hat{\sigma}}, \quad (54)$$

where $\hat{\mu}$ and $\hat{\sigma}$ are the sample mean and the sample variance of the errors. Z-scores are presented in figure 21, and it can be seen that there is a slight skew in the histogram when compared to the standard normal distribution. The histogram is not a perfect fit but it is still good in the sense that the Z-scores seem to be quite close to a bell curve. As stated earlier, the rain rates are log-normally distributed, hence it could be that the skew results from the reflectivity values that are not related to rainrates like hail or snow [16]. In total it seems that reflectivity can be modeled to some extent as a normal variable through SBL.

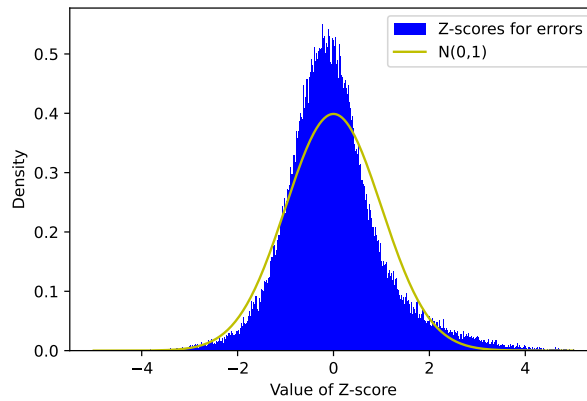


Figure 21: A histogram of the density of the Z-scores for the prediction errors to evaluate how well do the errors actually correspond to a standard distribution. The histogram seems to have most of its mass near the origin, but the values seem to have more density near the middle than for the standard normal distribution. There seems to also be positive skew in the distribution, more specifically the skew is about 0.74. A possible reason for the results differing from the standard normal distribution could be that only the reflectivities that imply rain are normally distributed, and other non-rain related scatterers, especially the ones in the melting layer, are distributed differently.

To demonstrate the interplay between the mean interpolation raster and the

standard deviation given by SBL, in figure 22 a histogram of the errors divided by the learnt standard deviation is presented. It is quite clear from the histogram that the predicted mean interpolation values mostly seem to fall quite close to the correct values from the test data when compared to the standard deviation from the model. This could mean that the standard deviation overestimates the magnitude of errors, but this can not be deduced for sure as the relationship between the errors and the learnt standard deviations is not explicit since the standard deviations model the uncertainty in the learnt model.

In addition to the difficulties with the convergence of the iteration, there were also problems related to attenuation correction and echo filtering. As attenuation correction is only derived for rain, the results might have changed a little if hail was removed from the observations. Also, the level of melting layer was specified with respect to the altitude from the sea level, while radar bins were presented in coordinates with respect to the antenna of the radar. This might have some effect on the results, but as the errors affect both the training and test sets in same ways, it could be that the results are not drastically different when the models are compared.

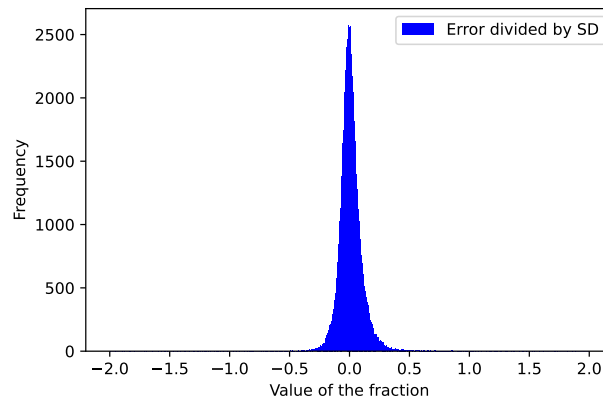


Figure 22: The histogram of errors obtained by the model divided by the computed standard deviation when interpolating to the large 256x256x64 raster. Values that differ from zero by less than one correspond to errors that deviate less than a computed standard deviation from the computed mean. As most of the fractions are less than one in absolute value, most error values are contained inside the standard deviations.

5.1 Ideas for improvements

As seen from the results there is clearly room for improvements if SBL is planned to be used for interpolation of radar volumes. More specifically, the method should be more efficient and more accurate overall. The problems in both of these cases are caused by the problems in matrix inversion by CG or by the diagonal estimation to

which CG is used. Hence, a natural first improvement idea would be to make the CG itself work more efficiently. To do this, one could test different preconditioners in order to get less ill-conditioned matrix to which CG is applied. The problem transforms to the problem of selection of memory efficient and well enough behaving preconditioning, which is not trivial.

A possible solution for the ill-conditioned matrices is to use a different form of basis where there is a smaller risk of severe ill-conditioning. A possible choice could be some Wavelet transform, but then the formulation of the parameters might be harder to be defined. Wavelet transforms might also make sense in terms of the operator pruning as there might be fewer bases with high covariances. This is due to the fact that some wavelets tend to have zero values outside the peak of the high-valued part, which would lead to smaller covariances with waves with peaks on elsewhere on raster. Maybe the approximation of matrix inverses would also work if the basis is different.

Another idea could be changing the computation of the posterior mean and the posterior variance to a method like Gaussian belief propagation. It would be a good way forward if the matrix is sparse, but for dense matrices the use might be too heavy. It is not entirely clear whether the usage of DCT as an operator for efficiency is possible for belief propagation, and hence this approach as itself might not work.

The inversion in (31) can be modified using the Woodbury identity so that the computation of the eigenvectors could be done efficiently using an offline learnt matrix \mathbf{B} for which $\mathbf{B}\mathbf{S}^T\mathbf{S}\Psi\mathbf{S}^T\mathbf{S} = \mathbf{S}^T\mathbf{C}\mathbf{S}\mathbf{S}^T\mathbf{S}$, where \mathbf{C} is a diagonal matrix that results from orthonormalization of \mathbf{B} . Here Ψ could be normalized DCT-IV which is its own inverse. This makes the matrix $\mathbf{S}\Psi\mathbf{S}^T$ invertible as a full rank matrix, making the existence of $\mathbf{S}\mathbf{C}\mathbf{S}^T$ possible. Then by normalizing \mathbf{B} , the eigenvalues can possibly be learnt from \mathbf{C} by scaling based on the normalization. The inversion would thus be straightforward, and possibly the badly behaving eigenvalues could be pruned away. \mathbf{B} could possibly be learnt using an approximation like neural network, which could be trained for given scanning strategy at a time.

An improvement for the considered approach could also be to set the starting values for σ^{-2} and $\boldsymbol{\alpha}$ to be some directly estimated values from an IDW or NN interpolated raster. Then the iteration would be based on sensible starting values. This could be done by giving an initial guess for σ^{-2} . Then the values for the prior precisions of the basis vectors could be estimated from the preinterpolated raster $\boldsymbol{\mu}_0$ using the formulas (31) and (32) as

$$\begin{aligned}\boldsymbol{\mu}_0 &= \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{t} \\ \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 &= \sigma^{-2}\boldsymbol{\Phi}^T\mathbf{t} \\ \mathbf{A}\boldsymbol{\mu}_0 &= \sigma^{-2}\boldsymbol{\Phi}^T(\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu}_0),\end{aligned}\tag{55}$$

where \mathbf{A} is easy to extract via element-wise division as it is a diagonal matrix. This would make sense as it takes time for the EM algorithm to get on the right track. A better initial approximation could make CG to converge correctly from the beginning of the iteration, hence yielding better results as EM might currently stuck iterating in saddle points of the likelihood.

Ideas from this approach could also be used to make the inversion more efficient using a multigrid approach. E.g., for a 256x256 raster, the raster could first be split in four parts of 128x128 rasters or possibly iteratively further to 64x64 raster, and so on. The values for the posterior mean $\boldsymbol{\mu}$ and the diagonal of $\boldsymbol{\Sigma}$ for current iteration could be first computed for the smaller rasters using operator probing. Then the larger raster could be handled as a pre-interpolated raster in similar fashion as in (55). This approach could possibly speed up CG as smaller rasters give better starting values, or it could enable application of a multigrid preconditioner for CG [52].

Multigrid approach could also be used to greedily first learn a representation on a coarse grid, fix basis on that level, and then learn a new set of basis vectors on a more fine-grained resolution, still keeping the earlier large level details fixed. This could be efficient and still give sensible results.

Another improvement could be to make more regular radar measurements that are not slices of same elevation and azimuth. However, this would mean changing the data sampling strategy of the radar during measurement. In any case, this could lead to basis vectors being sampled at points where there are smaller common fluctuations and hence smaller covariances. This would probably lead to better conditioned matrices as well. Also, interpolation into a non-cartesian coordinate system might have some effect for the covariances, and hence it could be a promising approach as well.

6 Summary

This work presents SBL for interpolation of radar volumes, and analyses its performance for a selected weather event compared to more popular interpolation methods from wradlib [9]. The method initially seemed promising as it had the benefit of generating an estimate of uncertainties in form of the standard deviations of the interpolation. Hence, the work was carried out in hope that the method could produce better interpolations and simultaneously give more information than earlier models used for radar volume interpolation.

First, the method based on Bayesian regression model, where the prior parameters were learnt using EM algorithm 1 to maximize the probability that the data corresponds to the model, was presented. A memory efficient approach was made possible by operator probing and CG described in [8] as well as by the use of type II discrete cosine transform for basis functions.

It was shown that the method learns a sensible presentation in the form of the mean interpolation raster and the standard deviation related to that raster. Even though the raster could be generated, the learning was quite slow and the iteration was not able to converge earlier than the set maximum number of iterations. The EM algorithm did not converge to the chosen threshold for any the rasters.

There were several problems formed due to the posterior precision matrix $\Sigma^{-1} = \sigma^{-2}\Phi^T\Phi + \mathbf{A}$ in (31) becoming ill-conditioned. This led to convergence issues with in the CG which realized as slow iterations. Another problem was due to the employed approximations in the operator probing (48), making it possible that the matrices become erroneously unsuitable for inversion by CG. These problems most likely made the method perform worse than the other methods tested as the number of iterations could not be scaled due to high time consumption. The method was also tested with only one radar volume due to issues with the computational time, and hence the results produced by the methods can not be considered statistically significant. Finally, the assumptions and behavior of the method were analysed and ideas for improvement were discussed.

The current implementation of the method is still too inefficient and inaccurate for practical needs of radar volume interpolation. Nevertheless, this work gives some hints on the direction for further endeavors to improve SBL, possibly leading to a better practical method for the problem at hand.

References

- [1] “Economic losses from climate-related extremes in Europe.” <https://www.eea.europa.eu/ims/economic-losses-from-climate-related>. Accessed: 2023-3-10.
- [2] “WMO: Weather.” <https://public.wmo.int/en/our-mandate/weather>. Accessed: 2023-3-10.
- [3] F. Fabry, *Fundamentals of weather radar measurements*, p. 8–31. Cambridge University Press, 2015.
- [4] J. Ritvanen, B. Harnist, M. Aldana, T. Mäkinen, and S. Pulkkinen, “Advection-free convolutional neural network for convective rainfall nowcasting,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 1654–1667, 2023.
- [5] Y. Qi and J. Zhang, “A physically based two-dimensional seamless reflectivity mosaic for radar qpe in the mrms system,” *Journal of Hydrometeorology*, vol. 18, no. 5, pp. 1327 – 1340, 2017.
- [6] R. A. Warren and A. Protat, “Should Interpolation of Radar Reflectivity be Performed in Z or dBZ?,” *Journal of Atmospheric and Oceanic Technology*, vol. 36, no. 6, pp. 1143 – 1156, 2019.
- [7] M. E. Tipping, “Sparse bayesian learning and the relevance vector machine,” *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001.
- [8] A. Lin, A. H. Song, B. Bilgic, and D. Ba, “Covariance-free sparse bayesian learning,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 3818–3831, 2022.
- [9] M. Heistermann, S. Jacobi, and T. Pfaff, “Technical note: An open source library for processing weather radar data (wradlib), hydrology 25 and earth system sciences, 17, 863–871,” 2013.
- [10] C. Bekas, E. Kokiopoulou, and Y. Saad, “An estimator for the diagonal of a matrix,” *Applied Numerical Mathematics*, vol. 57, pp. 1214–1229, 2007.
- [11] J. R. Shewchuk *et al.*, “An introduction to the conjugate gradient method without the agonizing pain,” 1994.
- [12] S. Dorn and T. A. Enßlin, “Stochastic determination of matrix determinants,” *Physical Review E*, vol. 92, no. 1, p. 013302, 2015.
- [13] J.-Y. Gu, A. Ryzhkov, P. Zhang, P. Neilley, M. Knight, B. Wolf, and D.-I. Lee, “Polarimetric attenuation correction in heavy rain at c band,” *Journal of Applied Meteorology and Climatology*, vol. 50, no. 1, pp. 39 – 58, 2011.

- [14] H. S. Park, A. V. Ryzhkov, D. S. Zrnić, and K.-E. Kim, “The hydrometeor classification algorithm for the polarimetric wrs-88d: Description and application to an mcs,” *Weather and Forecasting*, vol. 24, no. 3, pp. 730 – 748, 2009.
- [15] J. Probert-Jones, “The radar equation in meteorology,” *Quarterly Journal of the Royal Meteorological Society*, vol. 88, no. 378, pp. 485–495, 1962.
- [16] B. Kedem and L. S. Chiu, “On the lognormality of rain rate,” *Proceedings of the National Academy of Sciences*, vol. 84, no. 4, pp. 901–905, 1987.
- [17] “Bins, sweeps, and volumes.” https://iris.vaisala.com/doc/en_US/bins,_sweeps,_and_volumes.html. Accessed: 2023-3-10.
- [18] “JetStream Max: Volume Coverage Patterns (VCPs).” https://www.weather.gov/jetstream/vcp_max. Accessed: 2023-3-10.
- [19] R. E. Rinehart, *Radar for meteorologists*. University of North Dakota, Office of the President, 1991.
- [20] R. J. Doviak and D. S. Zrnić, “2 - electromagnetic waves and propagation,” in *Doppler Radar and Weather Observations (Second Edition)* (R. J. Doviak and D. S. Zrnić, eds.), pp. 10–29, San Diego: Academic Press, second edition ed., 1993.
- [21] D. E. Myers, “Spatial interpolation: an overview,” *Geoderma*, vol. 62, no. 1-3, pp. 17–28, 1994.
- [22] S. Park and M. Berenguer, “Adaptive reconstruction of radar reflectivity in clutter-contaminated areas by accounting for the space-time variability,” *Journal of Hydrology*, vol. 520, pp. 407–419, 2015.
- [23] J. Zhang, K. Howard, and J. Gourley, “Constructing three-dimensional multiple-radar reflectivity mosaics: Examples of convective storms and stratiform rain echoes,” *Journal of Atmospheric and Oceanic Technology*, vol. 22, no. 1, pp. 30–42, 2005.
- [24] V. Lakshmanan, “Image processing of weather radar reflectivity data: Should it be done in z or dbz,” *E-Journal of Severe Storms Meteorology*, vol. 7, no. 3, 2012.
- [25] A. Augst and M. Hagen, “Interpolation of operational radar data to a regular cartesian grid exemplified by munich’s airport radar configuration,” *Journal of Atmospheric and Oceanic Technology*, vol. 34, no. 3, pp. 495–510, 2017.
- [26] X. Zhang, J. He, Q. Zeng, and Z. Shi, “Weather radar echo super-resolution reconstruction based on nonlocal self-similarity sparse representation,” *Atmosphere*, vol. 10, no. 5, p. 254, 2019.

- [27] A. Geiss and J. C. Hardin, “Inpainting radar missing data regions with deep learning,” *Atmospheric Measurement Techniques*, vol. 14, no. 12, pp. 7729–7747, 2021.
- [28] E. Ruzanski and V. Chandrasekar, “Weather radar data interpolation using a kernel-based lagrangian nowcasting technique,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3073–3083, 2014.
- [29] L.-P. Wang, S. Ochoa-Rodríguez, J. Van Assel, R. D. Pina, M. Pessemier, S. Kroll, P. Willems, and C. Onof, “Enhancement of radar rainfall estimates for urban hydrology through optical flow temporal interpolation and Bayesian gauge-based adjustment,” *Journal of Hydrology*, vol. 531, pp. 408–426, 2015. Hydrologic Applications of Weather Radar.
- [30] S. L. Barnes, “A technique for maximizing details in numerical weather map analysis,” *Journal of Applied Meteorology and Climatology*, vol. 3, no. 4, pp. 396 – 409, 1964.
- [31] G. P. Cressman, “An operational objective analysis system,” *Monthly Weather Review*, vol. 87, no. 10, pp. 367–374, 1959.
- [32] T. Zhao and Y. Wang, “Statistical interpolation of spatially varying but sparsely measured 3d geo-data using compressive sensing and variational bayesian inference,” *Mathematical Geosciences*, vol. 53, no. 6, pp. 1171–1199, 2021.
- [33] H. Wang, W. Chen, Q. Zhang, X. Liu, S. Zu, and Y. Chen, “Fast dictionary learning for high-dimensional seismic reconstruction,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 7098–7108, 2020.
- [34] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature,” *Geoscientific model development*, vol. 7, no. 3, pp. 1247–1250, 2014.
- [35] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis*. Chapman and Hall/CRC, 1995.
- [36] D. J. MacKay, “Bayesian interpolation,” *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.
- [37] “Metropolitan and Micropolitan Statistical Area Population by Characteristics: 2020-2021.” <https://www.census.gov/data/datasets/time-series/demo/popest/2020s-metro-and-micro-statistical-areas-detail.html>. Accessed: 2023-3-27.
- [38] S. Pulkkinen, V. Chandrasekar, and T. Niemi, “Lagrangian integro-difference equation model for precipitation nowcasting,” *Journal of Atmospheric and Oceanic Technology*, vol. 38, no. 12, pp. 2125 – 2145, 2021.

- [39] “Next Generation Weather Radar (NEXRAD).” <https://www.ncei.noaa.gov/products/radar/next-generation-weather-radar>. Accessed: 2023-3-10.
- [40] “OpenStreetMap.” <https://www.openstreetmap.org/#map=10/32.7619/-97.1411>, copyright: openstreetmap.org/copyright. Accessed: 2023-3-23.
- [41] R. A. Houze Jr, “Mesoscale convective systems,” *Reviews of Geophysics*, vol. 42, no. 4, 2004.
- [42] “Rawinsonde (RAOB) Data Archive.” <https://mesonet.agron.iastate.edu/archive/raob/>. Accessed: 2023-3-10.
- [43] J. J. Helmus and S. M. Collis, “The python arm radar toolkit (py-art), a library for working with weather radar data in the python programming language,” *Journal of Open Research Software*, vol. 4, 2016.
- [44] S. S. Leroy, C. O. Ao, and O. Verkhoglyadova, “Mapping gps radio occultation data by bayesian interpolation,” *Journal of Atmospheric and Oceanic Technology*, vol. 29, no. 8, pp. 1062–1074, 2012.
- [45] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.
- [46] K. B. Petersen, M. S. Pedersen, *et al.*, “The matrix cookbook,” *Technical University of Denmark*, vol. 7, no. 15, p. 510, 2008.
- [47] S. Boussakta and H. O. Alshibami, “Fast algorithm for the 3-d dct-ii,” *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 992–1001, 2004.
- [48] J. Makhoul, “A fast cosine transform in one and two dimensions,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 1, pp. 27–34, 1980.
- [49] C. F. Caiafa and A. Cichocki, “Multidimensional compressed sensing and their applications,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 3, no. 6, pp. 355–380, 2013.
- [50] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform,” *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.
- [51] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [52] O. Tatebe, “The multigrid preconditioned conjugate gradient method,” in *NASA Langley Research Center, The Sixth Copper Mountain Conference on Multigrid Methods, Part 2*, 1993.