NTNU
Norwegian University of
Science and Technology

Master's thesis

2019

Scott William Christopher Bunting

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Mathematical Sciences

Scott William Christopher Bunting

Value of information analysis in the context of leakage detection in CO2 storage

July 2019

NTNU
Norwegian University of
Science and Technology

NTNU

Norwegian University of
Science and Technology

Value of information analysis in the context of leakage detection in CO2 storage

Scott William Christopher Bunting

Applied Physics and Mathematics
Submission date: July 2019
Supervisor: Jo Eidsvik, NTNU
Co-supervisor: Anouar Romdhane, SINTEF Industri

Norwegian University of Science and Technology
Department of Mathematical Sciences

Abstract

Carbon capture and storage (CCS) is seen as a promising strategy to reduce emissions of CO_2 into the atmosphere. Currently preparations are being done in Norway for a full-scale CCS project. CO_2 will be stored in deep geological formations, and one of the locations being studied for such storage is Smeaheia, located in the North Sea. One of the major risks related to such a storage project is leakage of CO_2, and it is important to design monitoring programs addressing this and other risks. Monitoring is expensive, so it is important to design monitoring programs in a smart way, in order to optimize the relationship between value and cost. One possible way to asses the value of a monitoring program is value of information analysis. In such an analysis one defines a decision problem and measures the value of information (VOI) as the additional value obtained by acquiring information before making the decision. This study concerns value of information analysis of seismic data in the context of CO_2 storage decisions. In particular we develop a framework to assess when in time a seismic survey has most value for leakage detection. The decision considered is whether or not to continue the injection of CO_2. In the framework, the simulation-regression approach is used to estimate the VOI at different times. This approach uses Monte Carlo simulation and statistical regression techniques to estimate the VOI. The framework is illustrated through a constructed case study using Smeaheia. Reservoir simulations are done using the Matlab Reservoir Simulation Toolbox (MRST). From simulated saturations of CO_2, we generate seismic data. We then regress values on the seismic data, to estimate the VOI. Two regression techniques are tested - k-nearest neighbors regression with principal components of the seismic data and convolutional neural networks. VOI estimates obtained using the k-nearest neighbors regressions were consistently lower than the estimates obtained using the convolutional neural networks. It is possible that one or both of the methods make biased estimates of the VOI. Through bootstrapping, we saw that the k-nearest neighbors approach produced stable VOI estimates, while the convolutional neural networks produced estimates with high variability. The high variability might be due to the limiting size of the data set. In the case study, we were not able to say exactly at which time a seismic survey would have the highest value. However, we were able to give a reduced interval of time in which the VOI would most probably obtain its highest value.

Sammendrag

Karbonfangst og -lagring er en lovende strategi for å redusere utslipp av CO_2 til atmosfæren og i disse tider forberedes et fullskala prosjekt i Norge. CO_2 vil bli injesert i dype geologiske formasjoner, og blant stedene som studeres for lagring er Smeaheia som ligger i Nordsjøen. En av hovedrisikoene forbundet med CO_2-lagring er lekkasje, og det er viktig å lage monitoreringsprogram som addresserer dette og andre risikoer. Monitorering er kostbart, så det er viktig å lage slike programmer på en smart måte som optimerer forholdet mellom verdi og kostnad. I et slikt arbeid bør en estimere verdien til et monitoreringsprogram. Dette kan gjøres ved å definere en beslutningssituasjon og så se på økningen i situasjonens verdi dersom en samler data før en tar beslutningen. Denne studien handler om å estimere verdien av informasjon for seismiske undersøkelser i forbindelse med CO_2-lagring. Et rammeverk utvikles for å finne ut ved hvilket tidspunkt en seismisk undersøkelse har størst verdi i forbindelse med å detektere CO_2-lekkasje. Beslutningen som studeres er om en skal fortsette eller avbryte lagringen av CO_2. I rammeverket brukes Monte Carlo simulering sammen med statistiske regresjonsteknikker for å beregne verdien av informasjon. Rammeverket illustreres gjennom et konstruert eksempelstudie knyttet til Smeaheia. Reservoarsimuleringer gjøres ved bruk av Matlab Reservoir Simulation Toolbox, og fra simulerte CO_2-metninger konstrueres seismiske data. Deretter brukes regresjonsmodeller for å beregne verdier gitt data, som så brukes for å estimere verdien av informasjon. To regresjonteknikker testes i studien, k-nærmeste naboer (KNN) og konvolusjonelle nevrale nettverk (CNN). Beregnede verdier av informasjon var konsistent lavere ved å bruke KNN sammenlignet med CNN. Det er mulig at estimatene fra en eller begge metodene ikke er forventningsrette. Videre, ved å gjøre bootstrapping, så kunne vi se at KNN produserer stabile estimater, mens CNN produserer estimater med høy varians. Den høye variansen kan skyldes den begrensede størrelsen på datasettet. I eksempelstudien fikk vi ikke til å oppgi et eksakt tidspunkt for når verdien av en seismisk undersøkelse vil være størst, men vi fikk til å oppgi et redusert tidsintervall hvor det er sannsynlig at den største verdien ligger.

Preface

This thesis is my work done in the course TMA4900 - Industrial Mathematics Master's Thesis, and is the conclusion of five years of study of Applied Physics and Mathematics at the Norwegian University of Science and Technology (NTNU) in Trondheim. Through the last parts of my studies, I have chosen to specialize in statistics.

I am truly grateful for the supervision provided by Jo Eidsvik at the department of Mathematical Sciences at NTNU and Anouar Romdhane at SINTEF Industry. Thanks for all the discussions, literature suggestions and feedback during the writing process. A special thanks to Per Bergmo, at SINTEF Industri, for joining discussions with Anouar and I and for sharing a reservoir model that was used in the early stages of this work. Thanks also to Halvor Møll Nilsen and Francesca Watson, at SINTEF Digital, for support regarding issues with the reservoir simulation. Through the work with this Master's Thesis, I have also contributed to the work of PhD student Susan Anyosa. She will present a poster on *A Simulation Analysis of CO2 Capture and Underground Storage Monitoring in Smeaheia* in the EAGE Conference on Petroleum Geostatistics, in Florence in September 2019. Thanks to Susan and Jo for letting me participate in this collaboration.

Scott William Christopher Bunting,
Trondheim, July 2019

Table of Contents

Chapter 1

Introduction

Nordbotten and Celia (2011) refer to the carbon problem as the ongoing increase of concentration of CO_2 in the atmosphere over the two last centuries. This increase is mainly driven by human activities (anthropogenic emissions) and most of the emissions come from the combustion of fossil fuels.

Figure 1.1 shows the Keeling curve, which consists of regular measurements of atmospheric concentration of CO_2 taken at Mauna Loa Observatory on Hawaii since 1958 (Keeling et al., 2001). It can be seen from the plot that current levels of CO_2 concentration in the atmosphere is above 400 ppm, while the concentrations in the late 1950s were at around 315 ppm. The concentrations fluctuate, but there has been a steady increase in concentration during the period that the measurements have been taken. In the introductory chapter of Nordbotten and Celia (2011), the Keeling curve is compared with measurements of atmospheric concentration of CO_2 from ice core data. Over the last 1000 years, ice core data show a stable level of CO_2 concentration at around 280 ppm and the increase above this level began with the industrial revolution. It is also stated that concentrations from ice core data over the past 650 000 years vary between about 170 ppm and 300 ppm. This comparison shows an extraordinary increase of atmospheric CO_2 concentration since the industrial revolution.

Higher levels of CO_2 concentration in the atmosphere contribute to an increased greenhouse effect and the consensus is that dangerous climate change is expected unless the increase is reduced or reversed (Nordbotten and Celia, 2011). The UN states that climate change is one of the major issues of our time and among the consequences are shifting weather patterns, which will threaten food production, and increasing sea levels, which bring increasing risks of catastrophic floods (United Nations, 2019).

Carbon capture and storage (CCS) has emerged as an option to reduce emissions of CO_2 into the atmosphere. It involves capturing produced CO_2, transporting it and storing it somewhere else than in the atmosphere. Possible storage locations are rock formations

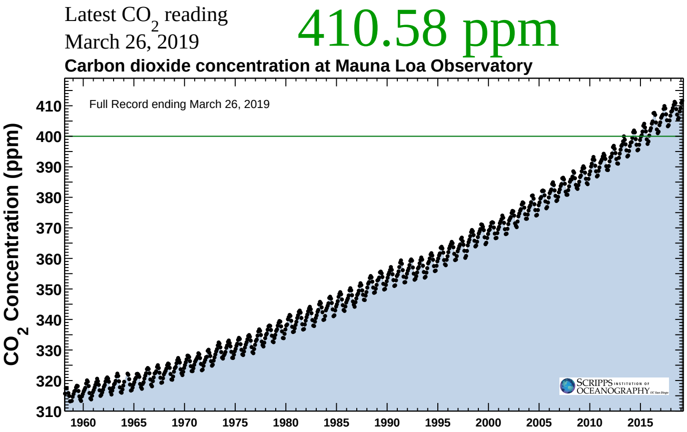


Figure 1.1: The Keeling curve: monthly measurements of atmospheric CO_2 concentration taken at the Mauna Loa Observatory on Hawaii (Keeling et al., 2001).

deep underground. In the Norwegian part of the North Sea, CCS has been done at the Sleipner gas field since 1996. Currently, other locations in the North Sea are being considered for a full-scale CCS project. Among those are Smeaheia, located approximately 58 km north west of Bergen.

Monitoring is an important part of CO_2 storage projects. In 2014 regulations were established in Norway, stating that monitoring programs need to ensure (Dupuy et al., 2018):

- Conformance: understanding of how the CO_2 behaves in the storage reservoir
- Containment: ensuring that the CO_2 migration is controlled
- Contingency: detecting and addressing leakages and other anomalies.

One of the main risks related to CO_2 storage is potential leakage, and it is important to design monitoring programs that detect these. Possible monitoring techniques include seismic and electromagnetic surveys.

At the Sleipner storage site seismic surveys have been conducted regularly. From the surveys, a large quantity of data has been made available for research, resulting in many published studies. See for instance Arts et al. (2004) and Dupuy et al. (2017). In Furre et al. (2017) 20 years of monitoring CO_2 injection at Sleipner is summarized. It is concluded that the monitoring program at Sleipner, which has strongly relied on seismic surveys, has been a success and that is has showed that the CO_2 has stayed safely in the storage unit.

Monitoring is costly, so it is important to design monitoring programs in a smart way that optimizes the relationship between value and cost. What technology to use, when to monitor, how often and to what extent, are all important questions to ask. Such considerations should be case specific and risk based (Furre et al., 2017).

One possible way to estimate the value of a monitoring scheme is value of information analysis (Howard, 1966). In such an analysis one assumes a decision problem and calculates the value of information (VOI) as the difference between the value of the decision problem with the information and the value of the decision problem without the information. The VOI is thus the additional information obtained by acquiring the information before making the decision.

In Eidsvik et al. (2015), VOI analysis is integrated with geostatistical modelling. The book provides applications to the petroleum industry, as well as mining and ground water management. Sato (2011) provides a high level introduction to VOI analysis in the context of CO_2 storage monitoring.

In real world applications, one is usually not able to calculate the VOI analytically. A computationally efficient approach to estimate the VOI in such cases, is the simulation-regression approach, presented and used in medical applications in Strong et al. (2014). This approach uses Monte Carlo sampling and statistical regression techniques to estimate the VOI. The approach has been integrated with geostatistics and applied to petroleum problems in Eidsvik et al. (2017) and Dutta et al. (2019).

The objective of the current study is to do VOI analysis of seismic data in the context of leakage detection in CO_2 storage. A decision problem is constructed where CO_2 is injected into a reservoir and the decision maker at some point will do a seismic survey and decide between continuing or stopping the injection, based on the result of the survey. Such a seismic survey will have varying VOI depending on when the survey is taken. In the study, the VOI will be estimated at different times, to see when it is most beneficial to do the survey. The analysis is done on a constructed case study with the Smeaheia storage site. Further we use the simulation-regression approach to estimate the VOI. We do a large number of simulations of CO_2 injections into the reservoir using the Matlab Reservoir Simulation Toolbox (MRST) (Lie, 2019). From these simulations we generate seismic data. We then train regression models, where values are regressed on the data, in order to obtain expected values given data, to use in the VOI estimation. For the regression part, we will try two different techniques: k-nearest neighbors with principal components of the seismic data and convolutional neural networks.

In chapter 2 the basics of CO_2 storage and monitoring is presented. This includes information about how to simulate CO_2 storage and seismic data. Chapter 3 contains a thorough introduction to VOI analysis together with examples that put it into a CO_2 storage context. The simulation-regression approach is also presented in chapter 3. The chapter ends with a presentation of how the simulation-regression approach will be applied in the case study, including an introduction to the regression techniques that will be tested, which are k-nearest neighbor (KNN) with principal components (PCs) of the data and convolutional neural networks. Chapter 4 contains the case study where we measure the VOI of seismic data at different times for a constructed decision problem relating to CO_2 storage at

Smeaheia. The chapter starts with an explanation of the workflow, followed by results. Chapter 5 concludes the study. In this study we use the programming languages Matlab and R.

Chapter 2

Basics of CO2 storage and monitoring

In this section we briefly explain the basics of CO_2 storage. We then give an introduction to reservoir simulation in MRST. Finally, we discuss seismic data.

2.1 CO2 storage

Deep geological formations are the most likely storage locations for CO_2.

Two important rock properties when evaluating the suitability of a potential storage location are *porosity* and *permeability*, which we now define:

- *Porosity*: The porosity ϕ of a medium is the proportion of the bulk volume which is occupied by void space. We thus have $0 \leq \phi \leq 1$ (Lie, 2019).
- *Permeability*: The permeability κ of a porous medium is its ability to transmit a single fluid, when the void space of the medium is completely filled by that fluid (Lie, 2019).

The relationship between porosity and permeability is not always straight forward, but in general, with other factors kept equal, an increase in porosity implies an increase in permeability. For a geological formation to be used for CO_2 storage, it would need to satisfy the following (Nordbotten and Celia, 2011):

1. have high permeability to accept a large quantity of CO_2,
2. be overlaid by low permeable rock to keep the CO_2 in place.

Among the formations being considered for storage are depleted hydrocarbon fields and saline aquifers. The latter are geological formations consisting of porous rocks saturated with salt water, commonly referred to as *brine*.

Once CO_2 has been captured, it gets transported to a storage location where it is compressed and injected into the subsurface. The common strategy is to inject CO_2 to a depth where the pressure and temperature exceed its critical point (Nordbotten and Celia, 2011). Thus, the CO_2 transitions into its subcritical phase. This phase is more dense than CO_2 in gas phase, which is ideal because the CO_2 then takes up less storage space and is less buoyant. The subcritical CO_2 inside a geological storage, is commonly referred to as a *plume*.

Before injection of CO_2, the saline aquifer is fully saturated with brine. When CO_2 is injected, it partially displaces the brine. The aquifer is then partially saturated with brine and CO_2. Because the CO_2 phase is less dense than the brine, the CO_2 will start to move upwards.

2.2 Reservoir simulation in MRST

The Matlab Reservoir Simulation Toolbox (MRST) (Lie, 2019) is an open source software implemented in Matlab for reservoir simulation and modelling. The toolbox offers simulation tools along with examples, but it is also possible to combine the data structures and methods of the toolbox to create your own simulations.

The toolbox has an add-on module called **co2lab**, which offers tools to simulate CO_2 injection and storage. In order to simulate CO_2 migration differential equations need to be solved. A discussion of these equations is outside the scope of this study. However, we note that **co2lab** uses vertical equilibrium models (SINTEF, 2016b). These assume that the fluid phases are in vertical equilibrium, which reduces the spatial dimension of the simulation. This offers superior speed compared to 3D simulations.

For the current study we will define an injection scenario by placing an injection well in a reservoir and setting injection rates and injection length. We will also switch between open and closed reservoir boundaries in order to control leakage of CO_2 and we will make perturbations to porosities and permeabilities. All this will be passed into one of the MRST solvers, and we will get as output vertically averaged CO_2 saturations for a 2D cell of the reservoir for different times.

2.3 Seismic data

Reflection seismic data is one way to estimate properties of the subsurface. One acquires such data by emitting sound waves into the subsurface and recording and processing the recorded echoes (Eidsvik et al., 2015). The reflected amplitudes depend on elastic properties of the subsurface, which again depend on lithology and fluids in the pore spaces.

To understand the relationship between seismic data and reservoir properties, one must use rock physics models. However, there is uncertainty related to such interpretations. The simplest form of seismic data, are reflections at normal incident angels, i.e. the source and the receiver are at the same location. In an effort to reduce uncertainty, it can be useful to collect and process reflection amplitudes at an increasing distance between the source and receiver. This gives reflection amplitudes for a range of different angles. Seismic data of this type is called amplitude-versus-offset (AVO) data. The acquisition of such data might not involve a different seismic survey than one used to just get zero-offset data, but the processing is different and could be more expensive (Eidsvik et al., 2015)

2.3.1 Elastic properties and AVO attributes

Assume a section of the subsurface with two different layers in the vertical direction. For instance, the top layer could be a cap rock, and the bottom layer could be an aquifer that is used for CO_2 storage. The layers have elastic properties listed below:

- V_p: p-wave velocity
- V_s: s-wave velocity
- ρ: bulk density.

We assume the elastic properties to be homogeneous in the individual layers. We call the top layer for layer 1 and the bottom layer for layer 2. The elastic properties for layer 1 we denote by V_{p1}, V_{s1} and ρ_1, and the elastic properties for layer 2 we denote by V_{p2}, V_{s2} and ρ_2. On the top of layer 1 is sea water, with elastic properties denoted by V_{p0}, V_{s0} and ρ_0.

Seismic waves are emitted from a source and the reflections are recorded by receivers. Both the source and receivers are at sea level connected to a moving boat. When the sound waves travel through interfaces with changing elastic properties, some of the energy will be reflected. The reflection amplitudes depend on the elastic properties at the two sides of the interface. The situation is illustrated in figure 2.1.

In what follows we will only discuss seismic data from the interface between layer 1 and 2, and we look at a specific point at this interface, say point A. Each receiver will receive a time series of seismic echoes. Processing is then done to distinguish between echoes from different points in the subsurface. We denote a processed echo by $R(\theta)$, where θ is the reflection angle. An approximate way to describe the relationship between R and θ is given by (Avseth et al., 2005)

$$R(\theta) \approx R_0 + G \sin^2 \theta, \tag{2.1}$$

where R_0 (intercept) and G (curvature) are AVO attributes which depend on elastic properties at the given point in the subsurface. Let $\Delta V_p = V_{p2} - V_{p1}$ and $V_{pm} = \frac{V_{p1}+V_{p2}}{2}$

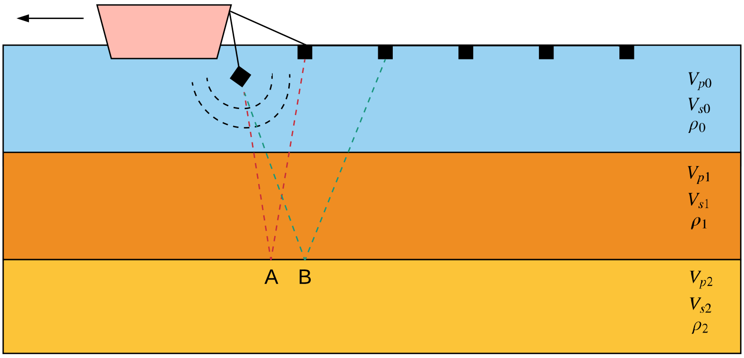


Figure 2.1: Illustration of a seismic survey.

(the arithmetic mean). We define similar quantities also for V_s and ρ. Approximate relationships between the AVO attributes and elastic properties are given by (Avseth et al., 2005)

$$R_0 = \frac{1}{2}\left(\frac{\Delta V_p}{V_{pm}} + \frac{\Delta \rho}{\rho_m}\right), \tag{2.2}$$

$$G = \frac{1}{2}\frac{\Delta V_p}{V_{pm}} - 2\left(\frac{V_s}{V_p}\right)^2\left(2\frac{\Delta V_s}{V_{sm}} + \frac{\Delta \rho}{\rho_m}\right). \tag{2.3}$$

By recording seismic amplitudes at different reflection angels, the AVO attributes for the given point can be estimated.

2.3.2 Rock physics models

In the current study, we want to use seismic data to map the CO_2 plume inside a reservoir. The elastic properties of a layer of rock, partially saturated with CO_2 and brine, can be approximated by rock physics relations, which will be described in this subsection.

The bulk density of a rock with fluid in its pore spaces is given by (Avseth et al., 2005)

$$\rho = \phi\rho_{\text{fluid}} + (1-\phi)\rho_{\text{mineral}}, \tag{2.4}$$

where ρ_{fluid} is the fluid density and ρ_{mineral} is the mineral density. The p-wave velocity depends on ρ, in addition to the rock bulk modulus K and the rock shear modulus μ. The s-wave velocity depends on ρ and μ. The relationships are given by (Avseth et al., 2005)

$$V_p = \sqrt{\frac{K + \frac{4}{3}\mu}{\rho}} \tag{2.5}$$

and

$$V_s = \sqrt{\frac{\mu}{\rho}}. \tag{2.6}$$

By relations from Gassmann, it is predicted that a change of rock fluid causes a change in value of K, but not a change in value of μ.

We now present a method to calculate the elastic properties of a rock partially saturated by CO_2 and brine. The method is taken from Avseth et al. (2005) and uses the Gassmann relations. It assumes that the elastic properties when the rock is saturated only by brine are known. We denote these initial values of the elastic properties by $V_p^{(1)}$, $V_s^{(1)}$ and $\rho^{(1)}$. The new values of the elastic properties, after CO_2 has partially replaced the brine, we denote by $V_p^{(2)}$, $V_s^{(2)}$ and $\rho^{(2)}$. The method consists of the following steps:

Step 1: Calculate the initial rock bulk modulus and rock shear modulus using equations (2.5) and (2.6):

$$\begin{aligned} K^{(1)} &= \rho^{(1)} \left((V_p^{(1)})^2 - \frac{4}{3}(V_s^{(1)})^2 \right) \\ \mu^{(1)} &= \rho^{(1)} \left((V_s^{(1)})^2 \right). \end{aligned}$$

Step 2: Use Gassmann's relation to calculate the new rock bulk modulus $K^{(2)}$. The relation is given by

$$\frac{K^{(2)}}{K_{\text{mineral}} - K^{(2)}} - \frac{K_{\text{fluid}}^{(2)}}{\phi(K_{\text{mineral}} - K_{\text{fluid}}^{(2)})} = \frac{K^{(1)}}{K_{\text{mineral}} - K^{(1)}} - \frac{K_{\text{fluid}}^{(1)}}{\phi(K_{\text{mineral}} - K_{\text{fluid}}^{(1)})},$$

where K_{fluid} is the fluid bulk modulus and K_{mineral} is the mineral bulk modulus. For the given problem we have $K_{\text{fluid}}^{(1)} = K_{\text{brine}}$, where K_{brine} is the bulk modules of the brine. $K_{\text{fluid}}^{(2)}$ can be calculated using the Brie average (Brie et al., 1995):

$$K_{\text{fluid}}^{(2)} = (K_{\text{brine}} - K_{CO_2})(1 - S)^e + K_{CO_2}.$$

Here, K_{CO_2} is the CO_2 bulk modulus and S is the CO_2 saturation. We set $e = 5$, as was done in Dupuy et al. (2017).

Step 3: Set the new rock shear modulus equal to the initial one:

$$\mu^{(2)} = \mu^{(1)}.$$

Step 4: Calculate the new bulk density $\rho^{(2)}$ using equation (2.4):

$$\rho^{(2)} = \phi\rho^{(2)}_{\text{fluid}} + (1-\phi)\rho_{\text{mineral}}.$$

$\rho^{(2)}_{\text{fluid}}$ can be calculated using the Voigt average (Dupuy et al., 2017):

$$\rho^{(2)}_{\text{fluid}} = (1-S)\rho_{\text{brine}} + S\rho_{\text{CO}_2},$$

where ρ_{brine} is the brine density and ρ_{CO_2} is the CO_2 density.

Step 5: Calculate the new p and s-wave velocities using equations (2.5) and (2.6) and the new values of K, μ and ρ:

$$V_p^{(2)} = \sqrt{\frac{K^{(2)} + \frac{4}{3}\mu^{(2)}}{\rho^{(2)}}}$$
$$V_s^{(2)} = \sqrt{\frac{\mu^{(2)}}{\rho^{(2)}}}.$$

2.3.3 Example

We now move on to illustrate the principles of the current section with an example. The goal is to see how well an AVO analysis is able to distinguish between different levels of CO_2 saturation in an aquifer. The example follows the workflow of the analysis in Dupuy et al. (2017).

Assume a reservoir as illustrated in figure 2.1. We assume layer 2 to be an aquifer used for CO_2 storage and layer 1 to be a cap rock. We will investigate five different cases of CO_2 saturation in layer 2: 0%, 25%, 50%, 75% and 100%.

We assume the following values for the elastic properties of layer 1:

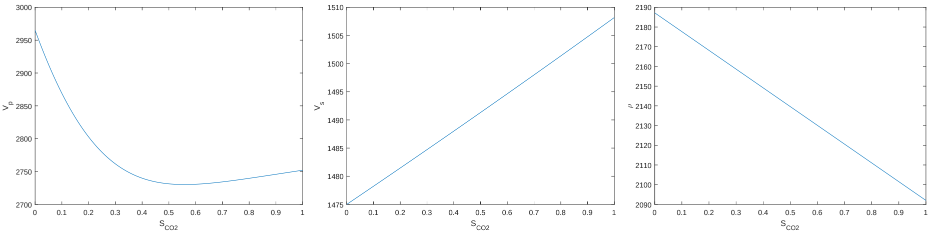
- $V_{p1} = 2546$ m/s
- $V_{s1} = 1115$ m/s
- $\rho_1 = 2278$ kg/m^3.

For layer 2, when it is fully saturated by brine, we assume the following values of the elastic properties:

- $V^{(1)}_{p2} = 2965$ m/s
- $V^{(1)}_{s2} = 1475$ m/s
- $\rho^{(1)}_2 = 2187$ kg/m^3.

The porosity for layer 2 we set to $\phi = 0.28$, and the mineral properties we set to:

- $K_{\text{mineral,2}} = 30.1$ GPa



(a) V_p **(b)** V_s **(c)** ρ

Figure 2.2: Elastic properties as a function of CO_2 saturation.

- $\rho_{\text{mineral},2} = 2.61$ kg/m^3.

Finally, the fluid properties used in the example are given in table 2.1.

Fluid phase	K_{fluid} (GPa)	ρ_{fluid} (kg/m^3)
Brine	2.308	1100
CO_2	0.075	760

Table 2.1: Fluid properties.

We start by calculating the values of the elastic properties with changing levels of CO_2 saturation. The calculations are done using the method presented in subsection 2.3.2 and the results are shown in figure 2.2. From the figure we see that the p-wave velocity changes rapidly among small levels of CO_2 saturation, while it changes less among larger levels. The s-wave velocity increases linearly with the saturation, while the density decreases linearly with the saturation.

The next step is to generate AVO attributes, R_0 and G, for the aquifer with changing levels of CO_2 saturation. We assume the processing of the AVO attributes to be noisy. Given the elastic properties we assume the AVO attributes to be normally distributed:

$$(R_0, G)^T \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{T}),$$

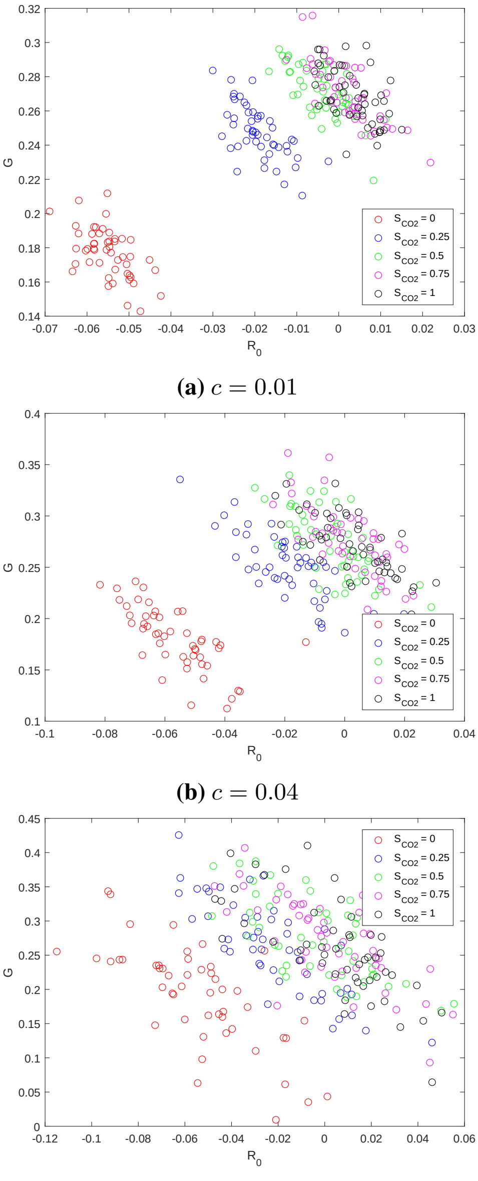
where the mean $\boldsymbol{m}$ is calculated using equations (2.2) and (2.3). The covariance matrix we set to

$$\boldsymbol{T} = c \begin{pmatrix} 0.06^2 & -0.7 \cdot 0.06 \cdot 0.17 \\ -0.7 \cdot 0.06 \cdot 0.17 & 0.17^2 \end{pmatrix},$$

for some $c > 0$. With $c = 1$, this covariance matrix corresponds to the one set for the likelihood model for AVO data in Eidsvik et al. (2015, p. 267).

We do our experiment with three different values of c: 0.01, 0.04 and 0.16. For each value of c, we generate 50 samples of AVO attributes for each of the five different levels of CO_2 saturation.

The results of the experiments are displayed in figure 2.3. For $c = 0.01$, we see from figure 2.3a that it is quite easy to discriminate between the saturations, except for the higher levels of saturations. As c increases, we see from figure 2.3b and 2.3c that the discrimination becomes more difficult. This behaviour is as expected, as more noise is added to the model with an increasing value of c.



(a) $c = 0.01$

(b) $c = 0.04$

(c) $c = 0.16$

Figure 2.3: 50 realizations of AVO attributes for five different levels of CO_2 saturation and different values of c. Note that the scale of the axes change between the plots.

Chapter 3

Value of Information Analysis and Statistical Methodology

In this chapter we look at decision problems under uncertainty and the value of acquiring additional information to inform the decision maker. The theory is introduced along with examples. In some cases the value of information (VOI) can be computed exact, while in others approximate methods are needed. We will look at both cases.

3.1 Decision analysis and value of information

Given is a decision maker which needs to choose some alternative $\boldsymbol{a} \in \boldsymbol{A}$, where $\boldsymbol{A}$ denotes the set of possible alternatives. Associated to the decision situation is some uncertainty $\boldsymbol{x}$. The value obtained by the decision maker after having selected $\boldsymbol{a}$ and observed $\boldsymbol{x}$, is given by $v(\boldsymbol{x}, \boldsymbol{a})$. The decision maker has some belief about the uncertainty $\boldsymbol{x}$, given by the prior probability distribution $p(\boldsymbol{x})$.

The following example is a slight modification of the motivating example in Sato (2011) and will be used as a running example to illustrate the theory in the coming subsections. A similar example can also be found in Eidsvik et al. (2018).

Example:
A company is considering CO_2 storage in a subsurface reservoir. The reservoir is intersected by a fault and it is uncertain whether CO_2 will leak across the fault or not. There is no other escape path for CO_2 from the reservoir.

The decision maker has two alternatives, $a \in \{0, 1\}$: to store CO_2 in the reservoir ($a = 1$) or not ($a = 0$). The uncertainty in this decision problem is binary, $x \in \{0, 1\}$, and is whether CO_2 will leak out from the reservoir ($x = 1$) or not ($x = 0$). The prior belief about the uncertainty is $p(x = 0) = 0.75$.

If the company chooses not to store the CO_2, they must pay 20 money units. The cost of storing the CO_2 is 10 money units. If CO_2 leaks out from the reservoir, the company must pay a fine of 32 money units. This gives the following value function $v(x, a)$:

- $v(0, 0) = -20$
- $v(0, 1) = -10$
- $v(1, 0) = -20$
- $v(1, 1) = -32 + (-10) = -42.$

3.1.1 Prior value

Assume a decision problem and a risk neutral decision maker. The optimal alternative is the one that maximizes the expected value (Eidsvik et al., 2015). Thus, the prior of the decision situation, before acquiring any additional information, is given by

$$PV = \max_{\boldsymbol{a}\in\boldsymbol{A}}\{\mathrm{E}[v(\boldsymbol{x}, \boldsymbol{a})]\} = \max_{\boldsymbol{a}\in\boldsymbol{A}}\left\{\int_{\boldsymbol{x}} v(\boldsymbol{x}, \boldsymbol{a})p(\boldsymbol{x})d\boldsymbol{x}\right\}. \tag{3.1}$$

We return to the CO_2 storage example. The expected values of the value function for the two different alternatives are:

$$\begin{aligned}\mathrm{E}[v(x, 0)] &= \sum_x v(x, 0)p(x) = v(0, 0)p(x = 0) + v(1, 0)p(x = 1)\\ &= -20 \cdot 0.75 + (-20) \cdot 0.25 = -20\end{aligned}$$

and

$$\begin{aligned}\mathrm{E}[v(x, 1)] &= \sum_x v(x, 1)p(x) = v(0, 1)p(x = 0) + v(1, 1)p(x = 1)\\ &= -10 \cdot 0.75 + (-42) \cdot 0.25 = -18.\end{aligned}$$

The prior choice is then to store CO_2 with $PV = -18$, calculated using equation (3.1). Notice that since x is a discrete variable, we use sums instead of integrals.

The decision situation is illustrated by the decision tree in figure 3.1. In such a figure, rectangles represent decisions and ovals represent uncertainties. The arcs out from the rectangles represent the different alternatives and the expected values of the value function when choosing the corresponding alternatives. The arcs out from the ovals represent the different outcomes of the uncertainties and their corresponding probabilities. The leafs of the tree represents the different outcomes of the decision situation and their corresponding values.

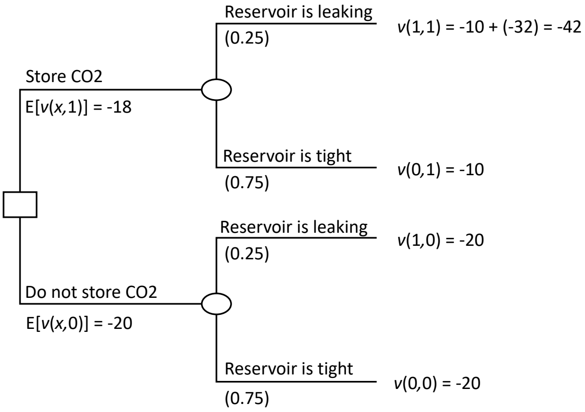


Figure 3.1: The decision tree for the CO_2 storage example without additional information.

3.1.2 Posterior value and value of information

Assume now that the decision maker can acquire some information, given by the uncertain variable $\boldsymbol{y}$. The dependence between $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by the likelihood $p(\boldsymbol{y}|\boldsymbol{x})$. The decision maker, has the opportunity to observe $\boldsymbol{y}$ before making the decision, and the posterior value of the decision situation, after having observed $\boldsymbol{y}$, is given by (Eidsvik et al., 2015)

$$PoV(\boldsymbol{y}) = \int_{\boldsymbol{y}} \max_{\boldsymbol{a}\in\boldsymbol{A}} \{\mathrm{E}[v(\boldsymbol{x},\boldsymbol{a})|\boldsymbol{y}]\} p(\boldsymbol{y}) d\boldsymbol{y}. \tag{3.2}$$

We further have

$$p(\boldsymbol{y}) = \int_{\boldsymbol{x}} p(\boldsymbol{y}|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \tag{3.3}$$

and

$$\mathrm{E}[v(\boldsymbol{x},\boldsymbol{a})|\boldsymbol{y}] = \int_{\boldsymbol{x}} v(\boldsymbol{x},\boldsymbol{a}) p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x}, \tag{3.4}$$

where $p(\boldsymbol{x}|\boldsymbol{y})$ is given by Bayes' theorem:

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}. \tag{3.5}$$

The VOI of $\boldsymbol{y}$, is given by the difference between the posterior value of the decision situation after having observed the information and the prior value before observing the information (Eidsvik et al., 2015). We have

$$VOI(\boldsymbol{y}) = PoV(\boldsymbol{y}) - PV. \tag{3.6}$$

The decision maker should acquire information $\boldsymbol{y}$ if $VOI(\boldsymbol{y})$ is higher than the price the decision maker has to pay to acquire the information.

A special case is when the decision maker is able to observe perfect information about $\boldsymbol{x}$ before making the decision. The posterior value in such a situation is (Eidsvik et al., 2015)

$$PoV(\boldsymbol{x}) = \int_{\boldsymbol{x}} \max_{\boldsymbol{a} \in \boldsymbol{A}} \{v(\boldsymbol{x}, \boldsymbol{a})\} p(\boldsymbol{x}) d\boldsymbol{x}, \tag{3.7}$$

and the value of perfect information is then

$$VOI(\boldsymbol{x}) = PoV(\boldsymbol{x}) - PV. \tag{3.8}$$

The value of perfect information is an upper bound for the value of information for a particular decision situation. If acquiring imperfect information $\boldsymbol{y}$ is more expensive than $VOI(\boldsymbol{x})$, the decision maker knows he should refrain from acquiring the information without even having calculated $VOI(\boldsymbol{y})$.

We again return to the CO_2 storage example. First we calculate the posterior value of the decision situation with perfect information about x:

$$\begin{aligned} PoV(x) &= \sum_{x} \max_{a \in A} \{v(x, a)\} p(x) \\ &= \max\{v(0,0), v(0,1)\} p(x=0) + \max\{v(1,0), v(1,1)\} p(x=1) \\ &= \max\{-20, -10\} \cdot 0.75 + \max\{-20, -42\} \cdot 0.25 = -12.5. \end{aligned}$$

The value of perfect information is then

$$VOI(x) = PoV(x) - PV = -12.5 - (-18) = 5.5.$$

Without doing further analysis, the company can conclude that they should not acquire any information about the given decision problem if the cost is larger than 5.5.

We now look at imperfect information. Assume that the company can perform a test indicating whether the reservoir is leaking or not. The result of the test is given by the random variable $y \in \{0, 1\}$, where the result $y = 1$ indicates that the reservoir is leaking and $y = 0$ indicates that it is sealing. The likelihood is given by $p(y = 0|x = 0) = p(y = 1|x = 1) = q = 0.9$.

We first calculate the probabilities for the different outcomes of the test:

$$\begin{aligned} p(y=0) &= p(y=0|x=0)p(x=0) + p(y=0|x=1)p(x=1) \\ &= 0.9 \cdot 0.75 + 0.1 \cdot 0.25 = 0.7 \end{aligned}$$

and

$$p(y=1) = 1 - p(y=0) = 0.3.$$

We then calculate the probabilities for the different outcomes of x, given the different outcomes of y:

- $p(x = 0|y = 0) = \frac{p(y=0|x=0)p(x=0)}{p(y=0)} = \frac{0.9 \cdot 0.75}{0.7} = 0.964$

- $p(x = 1|y = 0) = 1 - p(x = 0|y = 0) = 1 - 0.964 = 0.036$
- $p(x = 0|y = 1) = \frac{p(y=1|x=0)p(x=0)}{p(y=1)} = \frac{0.1 \cdot 0.75}{0.3} = 0.25$
- $p(x = 1|y = 1) = 1 - p(x = 0|y = 1) = 1 - 0.25 = 0.75.$

The expected values of the value function for the two different alternatives, given that the test reports $y = 0$, are

$$\begin{aligned} \mathrm{E}[v(x,0)|y=0] &= \sum_x v(x,0)p(x|y=0) \\ &= v(0,0)p(x=0|y=0) + v(1,0)p(x=1|y=0) \\ &= -20 \cdot 0.964 + (-20) \cdot 0.036 = -20 \end{aligned}$$

and

$$\begin{aligned} \mathrm{E}[v(x,1)|y=0] &= \sum_x v(x,1)p(x|y=0) \\ &= v(0,1)p(x=0|y=0) + v(1,1)p(x=1|y=0) \\ &= -10 \cdot 0.964 + (-42) \cdot 0.036 = -11.1, \end{aligned}$$

and the company would choose to store CO_2 ($a = 1$). Similarly, the expected values of the value function, given that the test reports $y = 1$, are

$$\mathrm{E}[v(x,0)|y=1] = -20$$

and

$$\mathrm{E}[v(x,1)|y=0] = -34,$$

and the company would choose not to store CO_2 ($a = 0$). This gives the posterior value

$$\begin{aligned} PoV(y) &= \sum_y \max_{a \in A}\{\mathrm{E}[v(x,a)|y]\}p(y) \\ &= \max_{a \in A}\{\mathrm{E}[v(x,a)|y=0]\}p(y=0) + \max_{a \in A}\{\mathrm{E}[v(x,a)|y=1]\}p(y=1) \\ &= -11.1 \cdot 0.7 + (-20) \cdot 0.3 = -13.8. \end{aligned}$$

We can then finally calculate the value of information:

$$VOI(y) = PoV(y) - PV = -13.8 - (-18) = 4.2.$$

The conclusion is then that the company should perform the test if the price is less than 4.2 money units.

The decision situation with imperfect information is illustrated by the decision tree in figure 3.2. The red ovals mark the optimal decisions given the two different outcomes of the test.

In figure 3.3 the value of information has been calculated and plotted for a range of different likelihoods. The value of information is 0 up until a certain threshold of the likelihood (just below 0.6). After this the VOI increases linearly with the likelihood up until the value of perfect information.

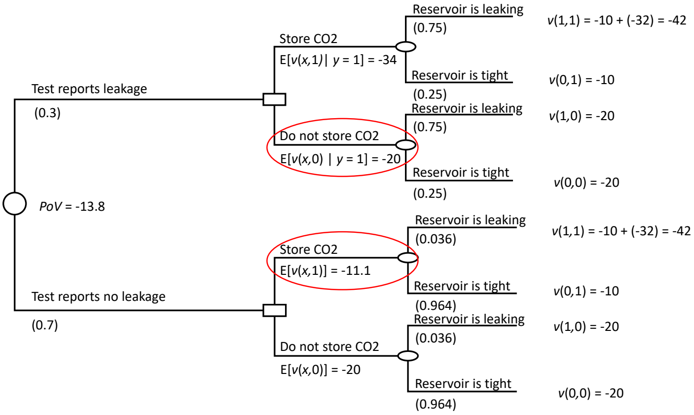


Figure 3.2: The decision tree for the CO$_2$ storage example with imperfect information.

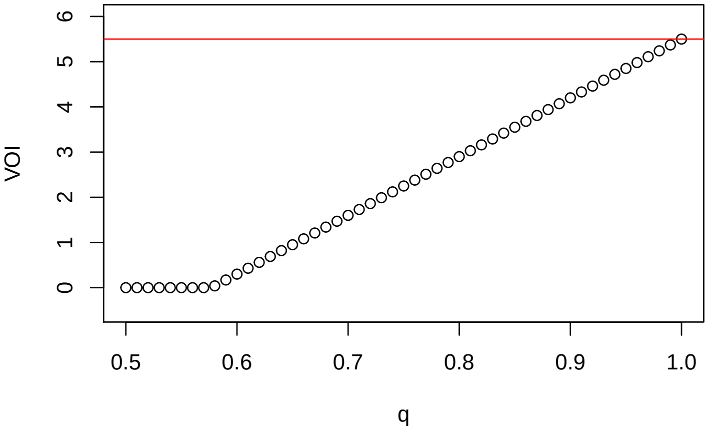


Figure 3.3: Value of information vs. likelihood q for the CO$_2$ storage example. The red line indicates the value of perfect information.

3.1.3 Conditions for valuable information

In order for information to be valuable it must fulfill three criteria; it must be relevant, it must be material and it must be economic (Eidsvik et al., 2015). The three different criteria are described further below:

1. Relevant: observing the information must have the capability to change the decision maker's belief about the uncertainty $\boldsymbol{x}$.

2. Material: observing the information must have the capability to change the decision of the the decision maker.
3. Economic: the value of information ($VOI(\boldsymbol{y})$) must be larger than the price to acquire the information.

The criteria can be illustrated by a pyramid as in figure 3.4. If the information is not relevant, it cannot be material, and if it is not material, it cannot be economic.

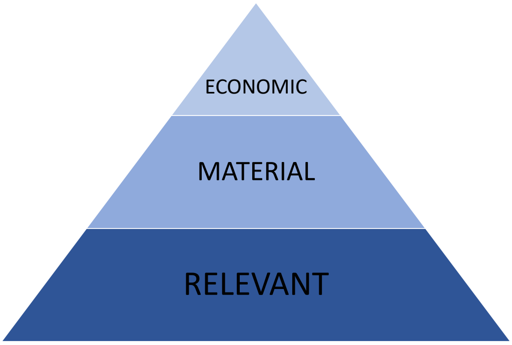


Figure 3.4: Criteria for information to be valuable.

We now look at the CO_2 storage example in light of these criteria. The information from the test is relevant. For instance, if the test reports $y = 0$, the company's belief about x changes from the prior probability $p(x = 0) = 0.75$ to the posterior probability $p(x = 0|y = 0) = 0.964$. Further, the information from the test is material. We have seen that if the test reports $y = 0$, the company will decide to store CO_2, while if the test reports $y = 1$, the company will decide not to store CO_2. Whether the information from the test fulfills the economic criteria, depends on the price of the test.

3.2 Time dependent value of information

In the case of CO_2 storage, it might be hard to detect if a fault will be leaking, without having injected any CO_2. In this section an extension of the CO_2 storage example is presented, where the accuracy of the test increases with time (and injected amount of CO_2).

Every year, for 10 years, a company will produce 1 unit of CO_2 that it wants to store in a subsurface reservoir. At the beginning of each year, the company will have the option to perform a test and to possibly stop the injection operation entirely. We assume that the test can only be performed once and that the injection can not be resumed once it has stopped. The main question is then: When is it most valuable to perform the test?

We denote the time in years by t. The start of the storage operation is at $t = 0$. The prior belief about x is $p(x = 0) = 0.75$, and we assume this is the best possible information available at $t = 0$. The accuracy of the test, as a function of time (assuming 1 unit of CO_2 has been injected into the reservoir each year up until that year), is $p(y = 0|x = 0, t) = p(y = 1|x = 1, t) = q(t) = 1 - 0.25 \cdot \mathrm{e}^{-\alpha t}$, where $\alpha > 0$.

The costs related to the operation are as follows:

- Fixed cost if injection is done: 5
- Cost of injecting per unit CO_2: 0.5
- Fixed cost if leakage: 2
- Fine if leakage per unit injected CO_2: 3
- Cost of not injecting per unit CO_2: 2.

Included in the fixed cost of injection is a mandatory test to detect leakage after year 9. We assume that this test will detect leakage with 100% accuracy. Thus, the company will for sure have to pay the fine if leakage occurs.

Given the available information at $t = 0$, the prior decision is to inject all units of CO_2, with $PV = -18$. We now go on to calculate the value of information from tests performed at years $t = 1, 2, ..., 9$, before injection is started for the given year. After the test, the company has the option to continue injecting ($a = 1$), or to stop entirely ($a = 0$). As mentioned, the company may only perform the test once, and thus the prior probability, will always be the one from $t = 0$.

The value function is now also a function of t, $v(x, a, t)$. If a test is performed at time t, the company will already have injected t units of CO_2, and the decision to make will be about the remaining $(10 - t)$ units. The posterior value will now be given by

$$
\begin{aligned}
&PoV(y,t) = -5 - 0.5t - (2 + 3t)p(x = 1)\\
&+ p(y = 0, t) \max\{-2(10 - t), -0.5(10 - t) - 3(10 - t)p(x = 1|y = 0, t)\}\\
&+ p(y = 1, t) \max\{-2(10 - t), -0.5(10 - t) - 3(10 - t)p(x = 1|y = 1, t)\},
\end{aligned}
$$

where the terms on the first line correspond to the expected value of the value function for the CO_2 already injected and the fixed costs. The two next lines correspond to the value of the CO_2 not already injected.

In figure 3.5 the value of information is plotted vs. the time the test is taken, with $\alpha = 0.2$. The value of information is largest when the test is done at $t = 4$, with $VOI(y, 4) = 1.24$.

In essence, the decision of when to perform the test in this case is a trade-off between test accuracy and amount of CO_2 put at risk of leakage.

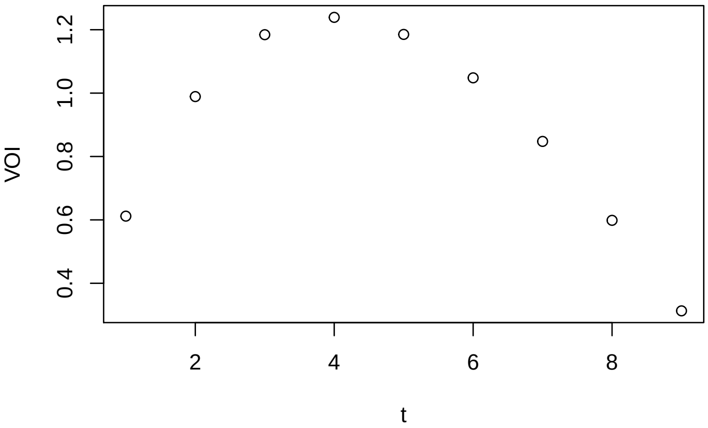


Figure 3.5: Value of information vs. time of test for CO_2 injection over time, with $\alpha = 0.2$.

3.3 Simulation-Regression Approach

When doing value of information analysis, there is often a need to approximate the posterior value given by equation (3.2). One possible way to do this is by the simulation-regression approach, which will be described in this section. The method is described and applied to petroleum geostatistics in Eidsvik et al. (2017). The approach presented in this section, will closely follow what is written in that article.

In practice, it is rare to have a closed form solution of the conditional expectation, $\mathrm{E}[v(\boldsymbol{x}, \boldsymbol{a})|\boldsymbol{y}]$, that appears in equation (3.2). The simulation-regression approach is based on Monte-Carlo sampling of data and then regressing the values on the data to find approximations of the conditional expectation. The approach consists of the following steps

1. Sample B realizations of $\boldsymbol{x}$, given by $\boldsymbol{x}^1, ..., \boldsymbol{x}^B$, from the prior probability $p(\boldsymbol{x})$.

2. For each realization, $\boldsymbol{x}^b$, of $\boldsymbol{x}$ and for each alternative $\boldsymbol{a} \in \boldsymbol{A}$, generate values $v_a^b = v(\boldsymbol{x}^b, \boldsymbol{a})$.

3. For each realization $\boldsymbol{x}^b$, generate data $\boldsymbol{y}^b$, using the likelihood $p(\boldsymbol{y}^b|\boldsymbol{x}^b)$ or some other relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$, $\boldsymbol{y}^b = f(\boldsymbol{x}^b)$, depending on what is available for the given case.

4. For each alternative $\boldsymbol{a}$, fit a model regressing the values on the data, $\hat{v}_a^b = g_a(\boldsymbol{y}^b, \hat{\boldsymbol{\beta}}_a)$, where $\hat{\boldsymbol{\beta}}_a$ are estimated model parameters. $\hat{v}_a^b$ is then an approximation of the conditional expectation $\mathrm{E}[v(\boldsymbol{x}, \boldsymbol{a})|\boldsymbol{y}^b]$.

5. Approximate the posterior value by

$$\begin{aligned} PoV(\boldsymbol{y}) = \int_{\boldsymbol{y}} \max_{\boldsymbol{a}\in\boldsymbol{A}} \{\mathrm{E}[v(\boldsymbol{x},\boldsymbol{a})|\boldsymbol{y}]\} p(\boldsymbol{y}) d\boldsymbol{y} &\approx \frac{1}{B}\sum_{b=1}^{B} \max_{\boldsymbol{a}\in\boldsymbol{A}} \{\mathrm{E}[v(\boldsymbol{x},\boldsymbol{a})|\boldsymbol{y}^b]\} \\ &\approx \frac{1}{B}\sum_{b=1}^{B} \max_{\boldsymbol{a}\in\boldsymbol{A}} \{g_a(\boldsymbol{y}^b, \hat{\boldsymbol{\beta}}_a)\} = PoV(\boldsymbol{y})_{\text{approx}}. \end{aligned} \tag{3.9}$$

6. Use $PoV(\boldsymbol{y})_{\text{approx}}$ to approximate the value of information.

It is considered good practice to also approximate the prior value by (Eidsvik et al., 2017)

$$PV_{\text{approx}} = \max_{\boldsymbol{a}\in\boldsymbol{A}} \left\{ \frac{1}{B}\sum_{b=1}^{B} g_a(\boldsymbol{y}^b, \hat{\boldsymbol{\beta}}_a) \right\}. \tag{3.10}$$

3.3.1 CO2 storage example

We will now test the simulation-regression approach on the CO_2 storage example from section 3.1.

The first step is to simulate B realizations of x, using the prior $p(x = 0) = 0.75$. From these realizations we use the likelihood to sample realizations of y. The values v_a^b for the different realizations of x and the different alternatives a, are set as in section 3.1. The data is binary and the conditional expectations are fitted as sample averages of the the values for the two different groups of data y.

We approximate the value of information 10 times for different values of B. The results are displayed in figure 3.6. We see that the approximations converge towards the true value of information with increasing B.

3.3.2 Time dependent VOI example

We now apply the simulation-regression approach to the time dependent VOI example from section 3.2 with $\alpha = 0.2$. We use $B = 5000$ and approximate the VOI 20 times for each of the years $1 - 9$. The results are displayed in figure 3.7. We see that on average the approximations are correct, but the variance in the estimates are large, especially for the early years.

3.4 Estimating the value of seismic data in CO2 storage monitoring

In this section we discuss how to estimate VOI for seismic data in CO_2 storage monitoring. The presented methodology builds upon the simulation-regression approach presented in

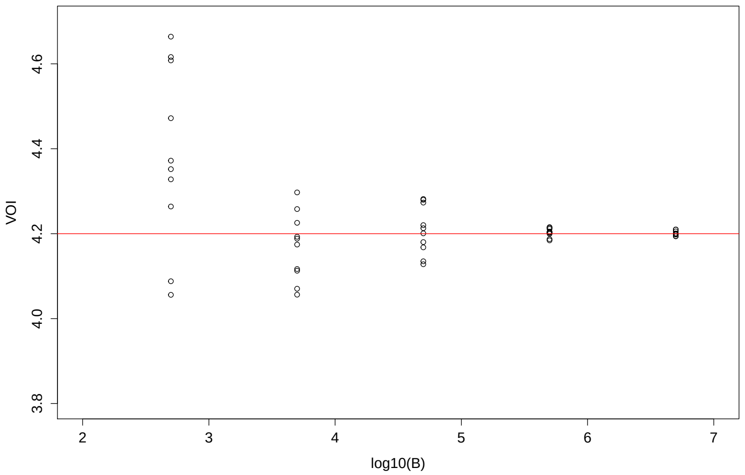


Figure 3.6: Approximate value of information vs. $\log_{10}(B)$, for the CO_2 storage example, using the simulation-regression approach. The red line indicates the true VOI.

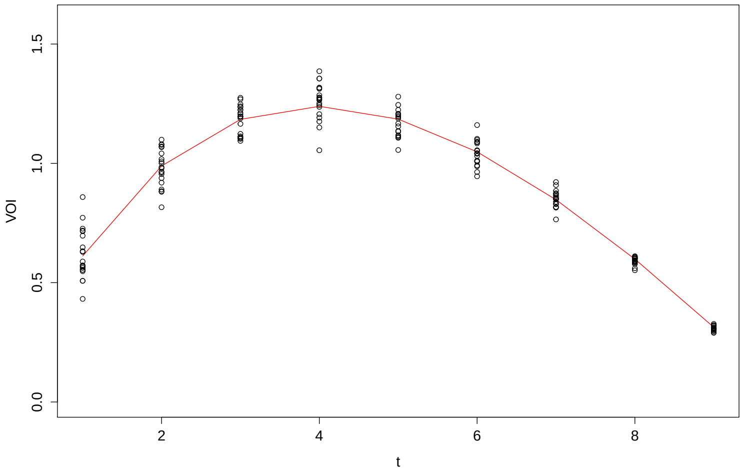


Figure 3.7: Approximate value of information vs. time, for the time dependent VOI example, using the simulation-regression approach with $B = 5000$. The red line indicates the true VOI.

section 3.3.

Assume a reservoir as in the previous sections, which may be leaking ($x = 0$) or not ($x = 1$). A company has injected CO_2 for a while, and is considering whether they should continue ($a = 1$) or stop ($a = 0$) the injection. They have the opportunity to collect and process seismic data in an attempt to see if the reservoir is leaking.

Seismic data, $\boldsymbol{y}$, would be acquired for a grid at the top of the reservoir, with dimension $I \times J$ cells. In the case of processing reflections at just normal incidence angels, one seismic attribute, R_0, would be available for each cell. In the case of AVO data, two seismic attributes, R_0 and G, would be available for each cell. In the first case $\boldsymbol{y}$ would be a vector of length IJ, and in the second case, $\boldsymbol{y}$ would be a vector of length $2IJ$.

In this case, we have no closed form solution for $\mathrm{E}[v(x, \boldsymbol{a})|\boldsymbol{y}]$, and some approximation method is needed to calculate the posterior value and VOI.

Building on the simulation-regression approach, presented in section 3.3, we suggest two different approaches to estimate the VOI. In both of the approaches, B Monte-Carlo samples of x and $\boldsymbol{y}$ are generated. We split these into two data sets. One data set of size B_1 to use for training of models and one data set of size B_2 to use for the estimation of VOI. We will estimate the posterior value using the approximation

$$PoV(\boldsymbol{y}) = \int_{\boldsymbol{y}} \max_{a \in A} \{\mathrm{E}[v(x,a)|\boldsymbol{y}]\} p(\boldsymbol{y}) d\boldsymbol{y} \approx \frac{1}{B_2} \sum_{b=1}^{B_2} \max_{a \in A} \{\mathrm{E}[v(x,a)|\boldsymbol{y}^b]\}, \tag{3.11}$$

inserted some approximation of the conditional expectation $\mathrm{E}[v(x,a)|\boldsymbol{y}^b]$. What differs between the approaches, is how we approximate $\mathrm{E}[v(x,a)|\boldsymbol{y}^b]$. The suggested approaches are:

1. Value regression with a nearest neighbors algorithm.
2. Estimate the conditional expectation by

$$\mathrm{E}[v(x,a)|\boldsymbol{y}^b] = \sum_x v(x,a) p(x|\boldsymbol{y}^b) \approx \sum_x v(x,a) \hat{p}(x|\boldsymbol{y}^b), \tag{3.12}$$

   where $\hat{p}(x|\boldsymbol{y}^b)$ is estimated by a convolutional neural network.

The approaches are described further below. We will also approximate the prior value by equation (3.10), in order to avoid the risk of getting negative VOI estimates.

3.4.1 Approach 1: Value regression by nearest neighbors averaging

In this subsection we describe the approach where the VOI is approximated by value regression with a k-nearest neighbors algorithm.

Given is a set of B Monte-Carlo samples of x, $\boldsymbol{y}$, $v(x,0)$ and $v(x,1)$. We keep B_1 realizations for training the algorithm, and set aside $B_2 = B - B_1$ realizations to approximate the VOI.

Let $\boldsymbol{y}^j$ be a realization of seismic data to which we want to approximate $v(x^j, a)$, for each a, and let $\mathcal{N}_0$ denote the set of k nearest neighbors to $\boldsymbol{y}^j$ in the training data. We then approximate the values by (James et al., 2013):

$$\hat{v}_a^j = \frac{1}{k} \sum_{i \in \mathcal{N}_0} v(x_i, a). \tag{3.13}$$

These approximations of the values are then used to approximate PV and PoV as in section 3.3.

Note that our vector of seismic data $\boldsymbol{y}$ is likely to be of high dimension. This can cause problems when using the nearest neighbors approach, due to curse of dimensionality. It might therefore be necessary to reduce the dimension of the seismic data by some dimension reduction technique. One such technique is principal component analysis (PCA), which will be described towards the end of this chapter.

3.4.2 Approach 2: Estimating the conditional probability

In this subsection we describe the approach where the VOI is approximated by estimation of the conditional probabilities $p(x|\boldsymbol{y})$.

We again have a set of B Monte-Carlo samples of outcomes and seismic data. We use B_1 of these samples to train a binary classification model. Such a model takes as input a realization of seismic data $\boldsymbol{y}$ and outputs an approximation of the conditional probability $p(x = 1|\boldsymbol{y})$. We denote the approximation by $\hat{p}(x = 1|\boldsymbol{y})$.

After having fitted the model using the training data, we use it to approximate conditional probabilities for the $B_2 = B - B_1$ observations of data in our hold-out set. The estimated probabilities are then used to calculate PV and PoV.

A popular approach for binary classification is to fit a neural network, which we will use in the current study. This method will be described in the end of this chapter.

3.5 Principal component analysis

In this section we describe principal component analysis (PCA), which can be used as a dimension reduction technique. The theory presented here is taken from James et al. (2013), where PCA is discussed as a dimension reduction technique before fitting a linear regression. The aim with this section is to give a brief explanation of what principal components are, and how you would use PCA in practice as a dimension reduction technique for regression.

Let Y be a data matrix of size $n \times p$, which consists of n observations of p variables. We denote a single observation of the p variables as $\boldsymbol{y} = (y_{i1}, y_{i2}, ..., y_{ip})^T$. When doing PCA, we first identify the direction in which our data varies the most. This direction is the first principal component direction. The second principal component direction is the direction among all the directions orthogonal to the first direction in which the observations vary the most, and so on. For an observation i, the jth principal component score is given as

$$z_{ij} = \phi_{1j}(y_{i1} - \bar{y}_1) + ... + \phi_{pj}(y_{ip} - \bar{y}_p), \tag{3.14}$$

with the restriction

$$\phi_{1j}^2 + ... + \phi_{pj}^2 = 1. \tag{3.15}$$

$\bar{y}_1$ here denotes the mean value of all the observations of the first variable, and so on. The principal component score is a linear combination of the original variables centered around zero and is the distance away from zero when the centered observation is projected on to the jth principal component line. The ϕs are called principal component loadings and describe the direction of the principal component. With n observations of p variables, it is possible to construct a maximum of $\min\{n-1, p\}$ principal components.

We illustrate the concept of principal components with an example. We draw 30 observations of 2 variables from a multivariate normal distribution $\mathcal{N}(\mu, \sigma)$, with

$$\mu = (45, 65)^T \quad \text{and} \quad \sigma^2 = \begin{pmatrix} 40 & 20 \\ 20 & 45 \end{pmatrix}.$$

The simulated observations, centered around zero, along with the two principal component directions, are illustrated in figure 3.8. We clearly see that the observations show large variability in the first principal component direction and that the directions are orthogonal to each other.

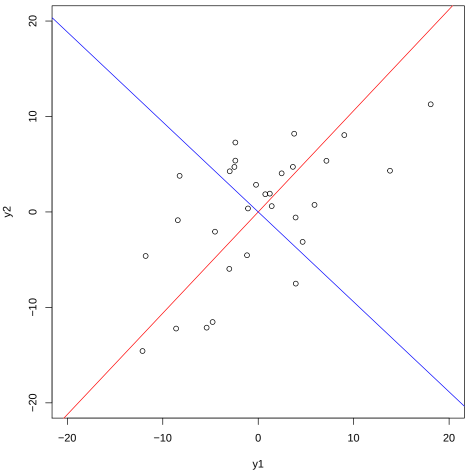


Figure 3.8: Simulated data, centered around zero, with the first principal component direction (red line) and second principal component direction (blue line).

When using principal components as a dimension reduction technique for a regression, the idea is to construct the first M principal components of the data and then use these as the covariates in the regression. The hope is that the components that explain most of the variability in the data, are also the ones that predict the target variable the best. In R, one can use the function **prcomp** to construct principal components. To decide the number of principal components to use in a regression, one can use cross-validation. It is in general advised to scale each set of variables to have unit variance before constructing principal components, especially if different variables are measured in different units. This is to make sure that all the variables are on the same scale.

3.6 Neural networks

In this section we describe neural networks applied to classification problems. We start by describing simple neural networks for classification, which consist of so called fully connected layers, and then move on to convolutional neural networks, which have a more complicated layer structure. Such networks are popular for image classification. Neural networks can be built in R, for instance using the Keras package (Allaire and Chollet, 2019). For this section, theory has been taken from Chollet and Allaire (2018).

3.6.1 Structure of a neural network

Assume a stochastic variable $x \in \{0, 1\}$ and data $\boldsymbol{y}$. For instance $\boldsymbol{y}$ could be the intensity of grey in each pixel of an image and x could be a variable describing whether the picture contains an apple or a pear. For a member in the population, the neural network will take as input $\boldsymbol{y}$ and return a prediction of the probability $p(x = 1|\boldsymbol{y})$.

Figure 3.9 illustrates the structure of a simple neural network with fully connected layers. The network takes as input a 3-dimensional vector $\boldsymbol{y}$, passes the input through a hidden layer, and outputs the prediction $\hat{p}(x = 1|\boldsymbol{y})$. In general, there can be more than one hidden layer in a neural network. Each layer consists of units. In our example, there are 3 units in the input layer, 2 units in the hidden layer and 1 unit in the output layer. The number of units in the input and output layer, correspond to the dimension of your input and output respectively, while the number of units in the hidden layer(s) must be decided when designing the network. The term *fully connected layers* refers to that each unit in a layer is connected to all the units in the next layer.

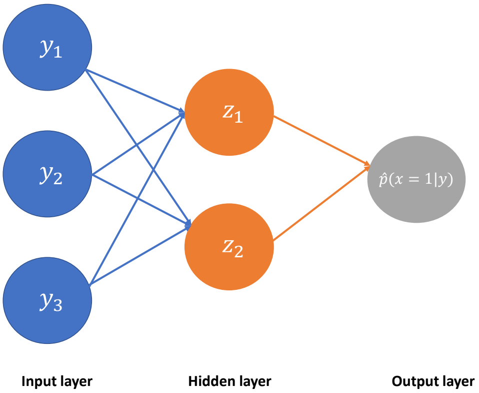


Figure 3.9: Illustration of the structure of a simple neural network with fully connected layers.

We will now describe how input is passed through a fully connected network to produce the output. The explanation will be based on the simple network in figure 3.9, but the generalization to more layers and units is straight forward.

The process starts by sending the input $\boldsymbol{y}$ from the input layer to the hidden layer. The first unit in the hidden layer takes as input each value from the input layer and outputs the number z_1. The second unit in the hidden layer also takes as input each value from the input layer and outputs the number z_2. An output value z_j from the hidden layer is computed in the following way:

$$z_j = f_1(\alpha_{1j}y_1 + \alpha_{2j}y_2 + \alpha_{3j}y_3 + \beta_j), \tag{3.16}$$

where f_1 is some activation function decided for the layer. The αs are model parameters. The output from the hidden layer is then passed on as input to the output layer where $\hat{p}(x = 1|\boldsymbol{y})$ is calculated in the same way as for the hidden layer, possibly with a different activation function.

What must be decided are model parameters for each layer. This is done through training on data with known values for x.

3.6.2 Activation function

When calculating output from a unit in a hidden layer it is common to use the activation function *rectified linear unit function* (relu) defined in the following way:

$$\text{relu}(y) = \max\{y, 0\}. \tag{3.17}$$

When the output in the output layer is an estimate of a single probability, we need the activation function to give us a number between 0 and 1. In such a situation it is common to use the *sigmoid function*:

$$\text{sigmoid}(z) = \frac{1}{1 + \mathrm{e}^{-z}}. \tag{3.18}$$

3.6.3 Training a neural network

Let $\boldsymbol{\gamma}$ be the set of model parameters. Further, let Y be a matrix whose rows are training data $\boldsymbol{y}_i$, $i = 1, ..., n$. Corresponding observations of the response are gathered in the $n \times 1$ vector X. When training a neural network one uses the training data and responses to find model parameters which make the network able to classify data with high accuracy.

The process of training a neural network is iterative. One starts with an initial guess of model paramters $\boldsymbol{\gamma_0}$ and then iteratively finds new parameters $\boldsymbol{\gamma}_k$ that improves the network. Central to the process is the *loss function*, which is a measure of how far away the predictions of the network are from the true values. When fitting a binary classification network, it is common to use a loss function called *binary cross-entropy*. Applied to our training observations, this function takes the form

$$l(\boldsymbol{\gamma}_k) = -\frac{1}{n}\sum_i^n x_i \text{log}(\hat{p}_k(x_i = 1|\boldsymbol{y}_i)) + (1 - x_i)\text{log}(1 - \hat{p}_k(x_i = 1|\boldsymbol{y}_i)), \tag{3.19}$$

where $\hat{p}_k(x_i = 1|\boldsymbol{y}_i)$ represents the predictions made by the network with current model parameters $\boldsymbol{\gamma}_k$. The goal in an iteration is to adjust the model parameters such that the loss function decreases. To do this one uses a method called *stochastic gradient descent*. This method updates the model parameters by moving their values in the opposite direction of the gradient of the loss function, $\nabla l(\boldsymbol{\gamma}_k)$, applied to some batch of the training data. For the current model parameters and the given batch, this is the direction that offers the largest decrease in the loss function. However, one must only make small adjustments to the model parameters, as one can not be sure that the loss function behaves as the gradient suggests outside the actual point $\boldsymbol{\gamma}_k$. Summarized, one iteration of the training process involves the following steps:

1. Make predictions for a batch of the training data using the current model parameters $\boldsymbol{\gamma}_k$.
2. Calculate the gradient of the loss function.
3. Update the model parameters by changing them a small quantity in the direction of the negative of the gradient. The formula for this update is:

$$\boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k + \delta \frac{\nabla l(\boldsymbol{\gamma}_k)}{|\nabla l(\boldsymbol{\gamma}_k)|}, \tag{3.20}$$

   where δ is some small step length.

Each observations of training data is placed into a batch. The size of the batches is a parameter that must be decided. The algorithm loops over the batches, until all of the training observations have had a change to update the model parameters. The process then repeats. One loop over all the training data is called an epoch.

One could let the training process go on until the performance of the network on the training data stops improving. However, one should be careful with overfitting. This happens when the network starts learning patterns that are just in the training observations and not in the general population of observations. This leads to a decrease of performance when predicting previously unseen data. To avoid overfitting one should hold out a set of observations from the training observations for validation. The training process should stop when the performance of the network starts to get worse on the validation set compared to the performance from previous epochs. Once the ideal number of epochs has been identified, one should train the network using all the observations in the training set.

3.6.4 Convolutional neural networks

Convolutional neural networks (CNNs) are popular for image classification. The main difference in architecture from a regular neural network consists of the introduction of a new kind of hidden layer called *convolutional layers*. The difference between a fully connected layer and a convolutional layer, is that fully connected layers learn global patterns in their input feature space, while convolutional layers learn local patterns in windows of the inputs.

For a regular neural network that we have discussed so far, the input feature map is given as a vector. For a CNN the input is given as a three dimensional tensor. Such a feature map is characterized by its width, height and depth, and the total number of cells is width × height × depth. For instance, if the input is a RGB picture, the width and height will be decided by the number of pixels in the picture, and the depth will be 3, one for each of the intensities of the colors red, green and blue. The depth is also commonly referred to as the number of channels.

A convolutional layer extracts patches from an input tensor, applies the same transformation to each patch, and places the transformed patches in an output tensor. Such a layer has two key parameters: the size (width and depth) of the patches to be extracted from the input, and the depth of the output. The depth of the extracted patches is the same as the depth of the input. The transformed patches have dimension $1\times$ output depth. The transformation of a patch consists of taking a dot product with a matrix of weights and applying an activation function. The weights will be the same for each patch and are the parameters that need to be learned for a convolutional layer. This transformation is done for every possible patch of the decided size that can be extracted from the input.

The transformation process of a convolutional layer is illustrated in figure 3.10. In this example the input tensor to the convolutional layer has (width, height, channels) = $(4, 4, 2)$, the patches are specified to have width and height both equal to 3 and the output depth is set to 3. This implies that each transformed patch will have dimension 1×3. There are four possible patches to be extracted from the input. The transformed patches get the same relative positions to each other in the output as the original patches had in the input. This means that the output width and height are both 2.

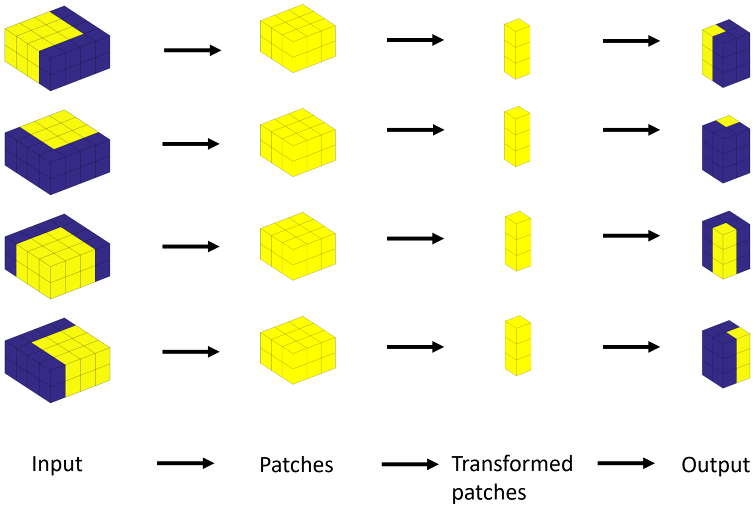


Figure 3.10: How a convolutional layer extracts patches from an input tensor, transforms them, and places the transformed patches in an output tensor.

An other common type of layer to include in CNNs are max pooling layers. The max pooling operation is a way to downsample feature maps. As with convolutional layers, a max pooling layer extracts patches from the input. We will discuss the most common case, which is to extract patches of size $2 \times 2\times$ input depth. When the algorithm goes through the input to extract the patches, it will move two cells at a time. This means that each cell will only appear in one patch. For each channel in each patch, the layer outputs the maximum value. In conclusion, the layer has downsampled the input feature map by a factor of 2 in the width and height dimension. One reason to do this, is to reduce the number of parameters that need to be learned in the network. Having too many parameters can easily lead to overfitting.

After having sent the features through some numbers of convolutional- and max pooling layers, it is common to also include one or more fully connected layers before the output layer. Before sending the current features into a fully connected layer, the feature map needs to be flattened into a vector.

3.6.5 R implementation

An implementation in R of neural networks can be found in the *keras* package. We here present a short example of code to build a CNN to do binary classification.

Assume we have 10000 grey scale images containing either apples or pears. We want to build a CNN to predict whether an image contains an apple or a pear. Each image has 28×28 pixels, each containing an intensity of grey.

The images are gathered in the variable **images** of size $10000 \times 28 \times 28 \times 1$. The images are split into **trainImages** and **testImages**, each of size $5000 \times 28 \times 28 \times 1$. Training will be done using **trainImages** and predictions will be done on **testImages**. The targets are gathered in the 10000×1 vector **targets**. If an image contains an apple, the corresponding target is 0. If an image contains a pear, the corresponding target is 1. As with the images, the targets are split into **trainTargets** and **testTargets**.

The CNN is designed with the following code:

```
library(keras)

model <- keras_model_sequential() %>%
  layer_conv_2d(filters = 32, kernel_size = c(3, 3),
        activation = "relu", input_shape = c(28, 28,1)) %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_conv_2d(filters = 64, kernel_size = c(3, 3),
        activation = "relu") %>%
  layer_flatten() %>%
  layer_dense(units = 64, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")

model %>% compile(
  optimizer = "rmsprop",
  loss = "binary_crossentropy",
)
```

The first hidden layer in the CNN is a convolutional layer. The output depth is defined by **filters** and the patch size is defined by **kernel_size**. In the first hidden layer, we also need to specify the size of the input feature map. In this example, each observation of the input has size $28 \times 28 \times 1$. The activation function is set by the variable **activation**. The next layer in the network is a max pooling layer. The patch size is here specified by **pool_size**. Following the max pooling layer is a new convolutional layer. After that, the features are flattende and passed through a fulle connected layer with 64 nodes, defined by **units**. The last layer is the output layer.

The last part of the code above consists of compilling the CNN. Here we specify that the loss function should be binary cross-entropy.

The next step is to train the network on the training data:

```
model %>% fit(trainImages, trainTargets, epochs = 50,
    batch_size = 64)
```

We here specify the batch size for each iteration of parameter changes to be 64. The number of epochs is set to 50.

With the trained model we can predict probabilities for the test images containing a pear:

```
prob <- model %>% predict_proba(testImages)
```

The probabilities found in the object **prob** can then be compared with the true targets.

Chapter 4

Case study

In this chapter we will simulate CO_2 storage and seismic data at Semaheia, a potential location for CO_2 storage in the Norwegian North Sea. Based on the simulations, we will estimate the value of information for seismic data in relation to detecting CO_2 leakage. The reservoir simulation is done with MRST, and the VOI is estimated using the methodology suggested in chapter 3.

The chapter starts with some background information about Smehaia. We then present the decision problem which is a variant of the time dependent VOI example in chapter 3. After that we explain the reservoir simulation and the simulation of seismic data in Matlab/MRST. Then we do the VOI calculations. The chapter ends with a discussion of the performance of the methods to calculate the VOI.

4.1 Smeaheia CO2 storage site

This section provides background material about Smeaheia, including geological information. The information is taken from Dupuy et al. (2018) and Norwegian Petroleum Directorate (2014).

Smeaheia is located in the North Sea within the Horda platform, approximately 58 km north west of Bergen in Norway. To the west of Smeaheia is the Troll gas field and the same reservoir formations are present in both, i.e. the Sognefjord, Krossfjord and Fensfjord formations. The storage site is a saline aquifer.

Smeaheia is bounded by two major faults, which are displacements in the rock. On the western side is the Vette fault and on the eastern side is the Øygarden fault complex. The situation is displayed in figure 4.1.

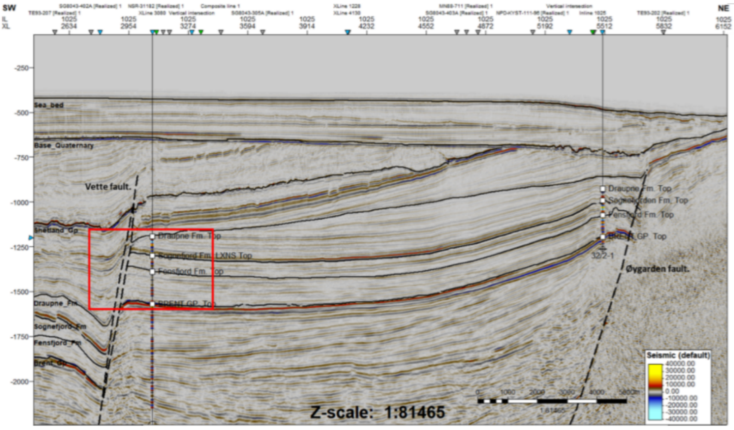


Figure 4.1: Smeaheia storage site. Taken from (Dupuy et al., 2018).

Sognefjord, Krossfjord and Fensfjord are the targets for CO_2 storage. They mainly consist of sandstone and siltstone in addition to some limestone stringers. The porosities are in the range between 25-32$\%$ and the permeabilities between 1-20 Darcy. On this basis the reservoir quality is expected to be good. On top of Sognefjord is the Heather formation, which consist of siltstone and claystone, and on top of Heather is the cap-rock seal of the Draupne formation, which consist of mudstone and claystone. Draupne is expected to have good sealing capacity, with porosities between 9-18$\%$ and very low permeabilities around $6 \cdot 10^{-9}$ Darcy. It is uncertain whether CO_2 will migrate through Heather or if it will be trapped beneath.

There are some uncertainties related to CO_2 storage at Smeahiea. One uncertainty is potential leakage through the faults. An other uncertainty is pressure communication with the Troll gas field. Currently further development at Smeaheia is put on hold and other locations are being considered for the CCS project. However, potential CO_2 storage at Smeaheia is still under consideration.

4.2 Decision problem

In this section the decision problem for the case study is presented. The decision problem is similar to the one that was used when studying time dependent VOI in chapter 3. The main point here is to decide when a seismic survey has highest value when it comes to detecting a potential leakage of CO_2.

Assume a company that is producing CO_2 and is going to start a CO_2 storage operation at Smeaheia. They want to inject 8 megatons of CO_2 each year for 25 years. There is an uncertainty $x \in \{0, 1\}$, relating to whether the reservoir is leaking ($x = 1$) or not ($x = 0$).

We let $t \in \{0, 1, ..., 24\}$ denote the time in years. The best available information at $t = 0$, right before the injection of the first unit of CO_2, says $p(x = 0) = 0.75$. This is the prior belief. The company may perform a seismic survey during the injection. Our task is to calculate the value of such a survey done at different times, assuming the company will only do the survey once.

The costs in this problem are as follows:

- Fixed cost if injection is done: 5
- Cost of injecting per unit CO_2: 0.2
- Fixed cost if leakage: 2
- Fine if leakage per unit of injected CO_2: 1.2
- Cost of not injecting per unit CO_2: 0.8.

We assume there is a mandatory survey done after the 25 years, at a time where one knows for certain whether CO_2 has leaked or not.

At time t the decision maker has two alternatives $a \in \{0, 1\}$.

- stop injection at time t ($a = 0$)
- continue injection ($a = 1$)

Assuming injection has been done every year until time t, the company will have injected t units of CO_2 at this time. The values, $v(x, a)$ for the different alternatives and outcomes are then:

- $v(0, 0) = -5 - 0.2t - 0.8(25 - t)$
- $v(1, 0) = -5 - 0.2t - 0.8(25 - t) - 2 - 1.2t$
- $v(0, 1) = -10$
- $v(1, 1) = -42$.

Without any seismic survey, the prior optimal alternative will be to continue injection for all times t, with $PV = -18$.

4.3 Workflow

Figure 4.2 illustrates the workflow of the case study. The process starts with 10.000 reservoir simulations. To each simulation we input a reservoir model of the Sognefjord formation. The porosities and permeabilities are permuted, so they will be different for each simulation. According to the prior probability of x, we draw if each simulation will be a leakage ($x = 1$) or no leakage ($x = 0$) case. If it is a leakage case, the boundaries of the reservoir will be set to open. If it is a no leakage case, the boundaries will be set to closed. The output of the simulations are 10.000 instances of CO_2 saturations. From these, we generate seismic AVO data in a grid which represents the area of interest inside the reservoir. We split the dataset into two. The first part of the dataset we use, along with the known values of x, to train statistical models. We then use these models to calculate the VOI on the second part of the data.

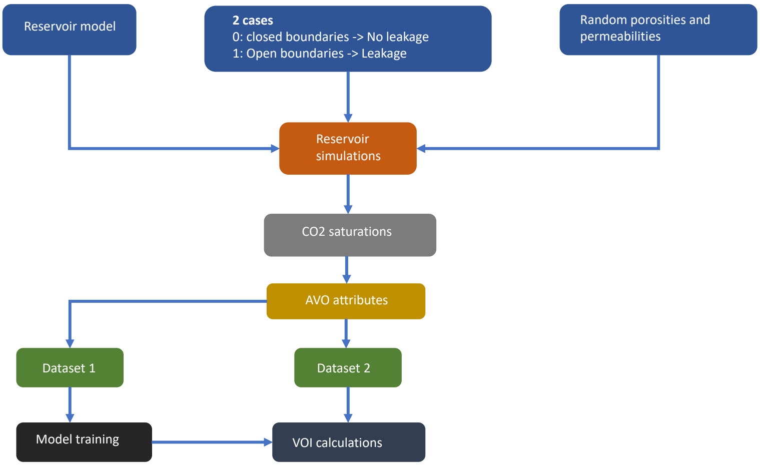


Figure 4.2: Workflow of the case study.

4.4 Reservoir simulation in MRST

The reservoir simulation is done using MRST. The simulation script for the case study strongly builds on the example *Fully-implicit VE simulation*, which can be found on the MRST website (SINTEF, 2016a). In that example VE simulation is done using a top surface grid of the Gassum formation. In our simulation this grid is replaced with a top surface grid of the Sognefjord formation. We place the injection well in the eastern part of the formation (see figure 4.4).

For each simulation we inject 8 Mt of CO_2 each year for 25 years. After the injection we simulate migration of CO_2 in the reservoir for a period of 25 further years. For each simulation we draw the scenario, leakage ($x = 1$) or no leakage ($x = 1$). If a simulation is drawn to be a leakage scenario, we set the boundaries to open. This will lead to a considerable amount of CO_2 leakage through the eastern boundary. If a simulation is drawn to be a no leakage scenario, the reservoir boundaries are set to be closed. In these cases there will not be any significant amount of leakage. All the simulations are run with injection for 25 years. For open boundary cases, if injection were to stop at an earlier time than after 25 years, we can not be sure if CO_2 will always leak. However, in the current case study we make this assumption.

Random porosities are generated for each cell for each simulation run. The porosities are specified to be a fixed mean, plus a random component which is spatially correlated according to the lateral coordinates of the cells. The mean value is specified to be 0.28 and we also specify that the porosities have to take values in the range $[0.26, 0.30]$.

From the simulated porosity ϕ in a cell, we generated the corresponding permeability (in millidarcy), κ, using the deterministic formula

$$\kappa = 0.000029 \cdot \mathrm{e}^{60.346\phi}. \tag{4.1}$$

This relationship was found by regression using log and core data and was provided by Gassnova. The allowed interval of porosities give permeabilities in the interval $[188.00\text{ mD}, 2112.4\text{ mD}]$. Figure 4.3 shows a plot of permeability as a function of porosity.

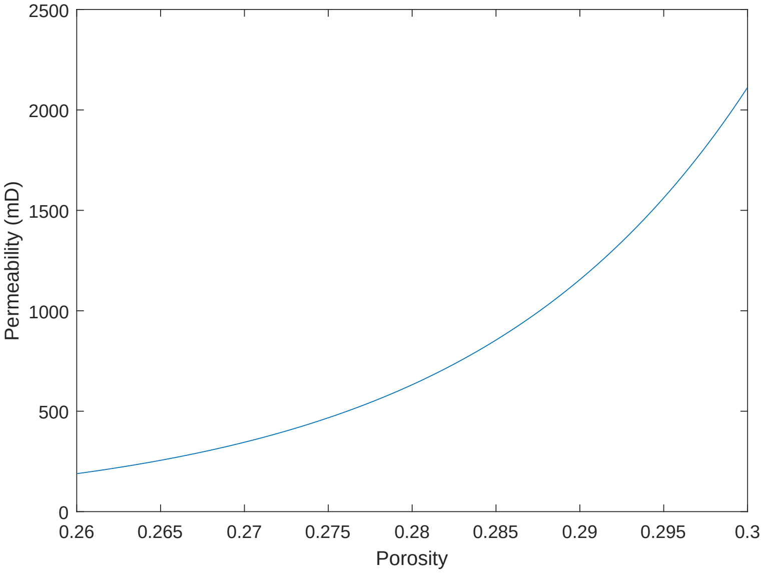


Figure 4.3: Modelled relationship between porosity and permeability.

For each simulation we note the value of x and extract saturations for cells in a particular area of the reservoir for different times. This area is marked in figure 4.4 and contains the injection well and lays next to the eastern boundary. The dimension of the area is 18×18. The reported saturations for a cell are vertically averaged saturations over the reservoir thickness for the given cell. We assume the CO_2 to rise to the top of the reservoir and the porosity to be constant for the whole depth of a cell. With these assumptions, if the reported saturation for a cell is 0.5, the top half of the reservoir of that location will be fully saturated by CO_2. We are interested in the saturations at the top of the reservoir and will use the assumed saturation at the top 5 meters of each cell in the grid for the AVO data simulation. For a cell, with average saturation S_{av} and thickness z, we estimate the

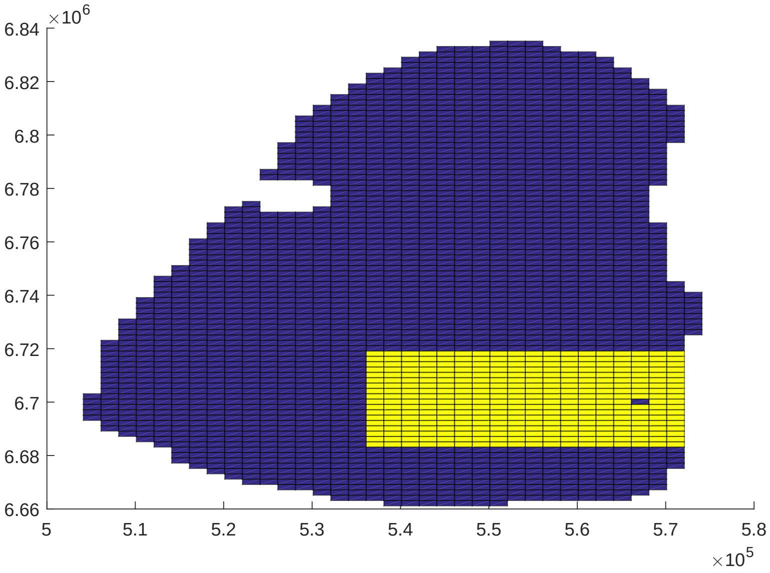


Figure 4.4: Reservoir grid, with the area of the seismic survey marked in yellow. The well is placed in the blue cell inside the yellow area (but seismic data is also collected there).

saturation S at the top 5 meters by the formula:

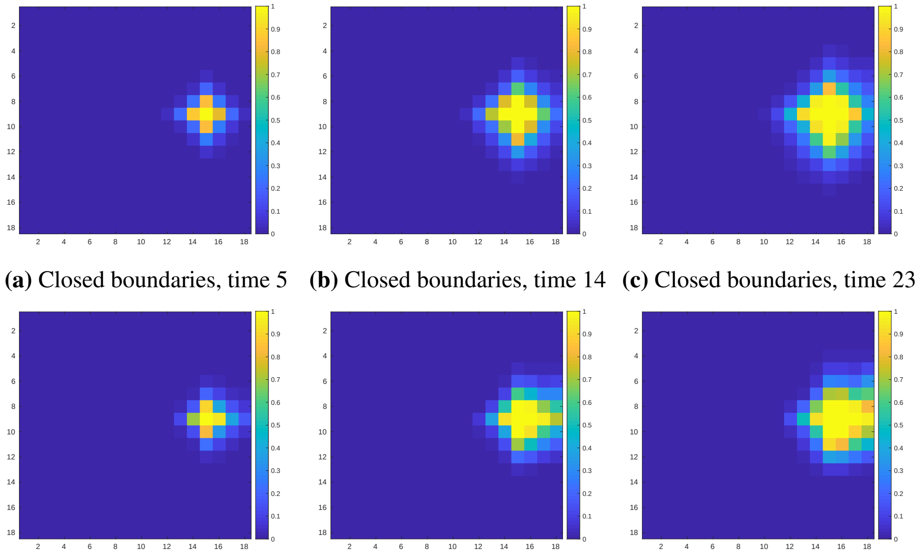
$$S \approx \frac{\min\{S_{\text{av}} \cdot z, 5\}}{5}.$$

We will estimate VOI for 8 different time points, $t \in \{2, 5, 8, 11, 14, 17, 20, 23\}$, and thus export and transform saturations from each simulation for these times. Figure 4.5 shows the average saturation for the two different scenarios of boundary conditions at different times. We see that on average, the CO_2 plumes behave differently depending on the boundaries. In figure 4.6 saturations for the same times have been plotted for one run of simulations for each of the different boundary cases. We see that the CO_2 plume behaves differently for these two simulations. Notice, however, that the boundary conditions are not the only thing that differs between the simulations - they also have different porosities and permeabilities, which also contributes to different plume behaviours.

4.5 AVO data

To estimate elastic properties and simulate AVO data, we follow what was presented in section 2.3, including the workflow of the example in subsection 2.3.3.

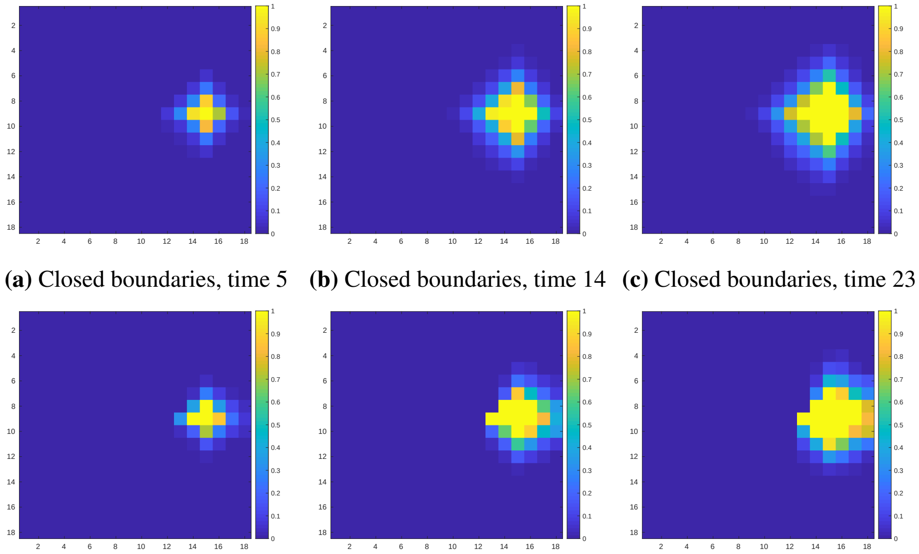
To estimate elastic properties of the cap rock and initial p- and s-wave velocities of the top of the reservoir, we use well-log data from Dupuy et al. (2018). The elastic properties of



(a) Closed boundaries, time 5 **(b)** Closed boundaries, time 14 **(c)** Closed boundaries, time 23

(d) Open boundaries, time 5 **(e)** Open boundaries, time 14 **(f)** Open boundaries, time 23

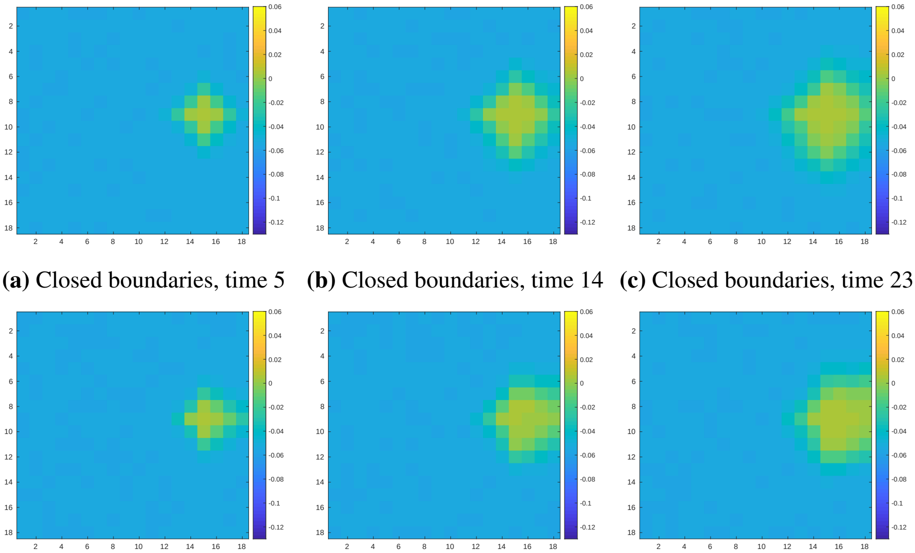
Figure 4.5: Average saturations for the two different values of x at times 5, 14 and 23.



(a) Closed boundaries, time 5 **(b)** Closed boundaries, time 14 **(c)** Closed boundaries, time 23

(d) Open boundaries, time 5 **(e)** Open boundaries, time 14 **(f)** Open boundaries, time 23

Figure 4.6: Saturations for one simulation run for each value of x at times 5, 14 and 23.



(a) Closed boundaries, time 5 **(b)** Closed boundaries, time 14 **(c)** Closed boundaries, time 23

(d) Open boundaries, time 5 **(e)** Open boundaries, time 14 **(f)** Open boundaries, time 23

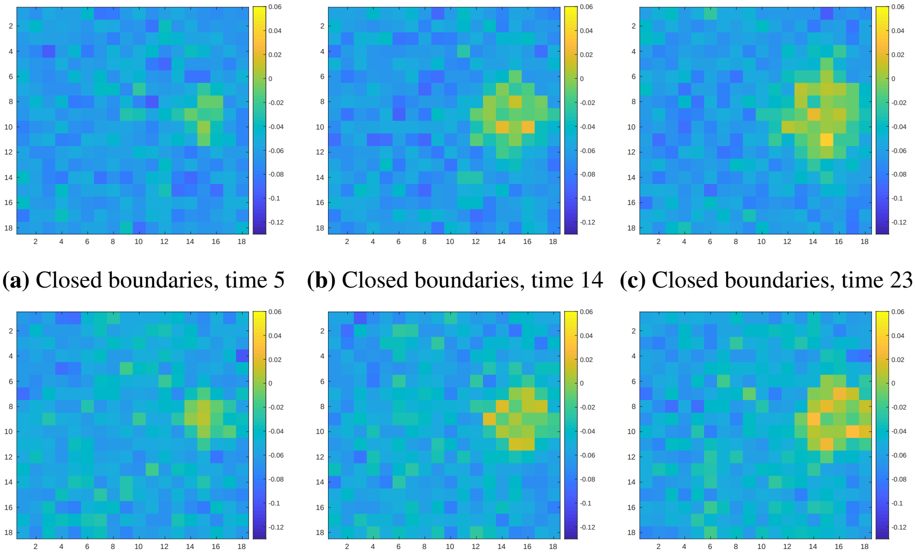
Figure 4.7: Average R_0 attributes for the two different values of x at times 5, 14 and 23.

the cap rock are estimated by taking the average value of the properties over a thickness of 65 meters above the interface between Heather and Sognefjord formations. The initial velocities of the top of the reservoir are estimated by taking the average value of the velocities over a thickness of 65 meters into the reservoir. The estimated values correspond to the values used in the example in subsection 2.3.3. The initial bulk density of the reservoir we calculate for the different cells with varying porosities using equation (2.4). The initial velocities of the reservoir probably also depend on the porosity, but we assume these to be constant for our rang of changing porosities. The fluid and mineral properties are taken from Dupuy et al. (2018), excpet for the fluid densities, which are taken from the fluid object that was calculated in MRST for the simulatuions. The fluid properties are the same as those used in the example.

To simulate the noise in the AVO data for each cell, we use the same covariance matrix, $\boldsymbol{T}$, as in subsection 2.3.3, with $c = 0.04$. The noise in different cells and from different times are assumed to be independent.

Figure 4.7 shows average R_0 attributes for the two different scenarios of boundary conditions at different times. Comparing with figure 4.5, we see that the average of R_0 attributes give a good picture of the average saturations. Figure 4.8 shows R_0 attributes for the same times for one run of simulations for each of the different boundary cases. Comparing with figure 4.6, which contains the same simulation runs, we see that the R_0 attributes give quite a good picture of the saturations, but there is quite a lot of noise.

Figure 4.9 shows average G attributes for the two different scenarios of boundary condi-



(a) Closed boundaries, time 5 **(b)** Closed boundaries, time 14 **(c)** Closed boundaries, time 23

(d) Open boundaries, time 5 **(e)** Open boundaries, time 14 **(f)** Open boundaries, time 23

Figure 4.8: R_0 attributes for one simulation run for each value of x at times 5, 14 and 23.

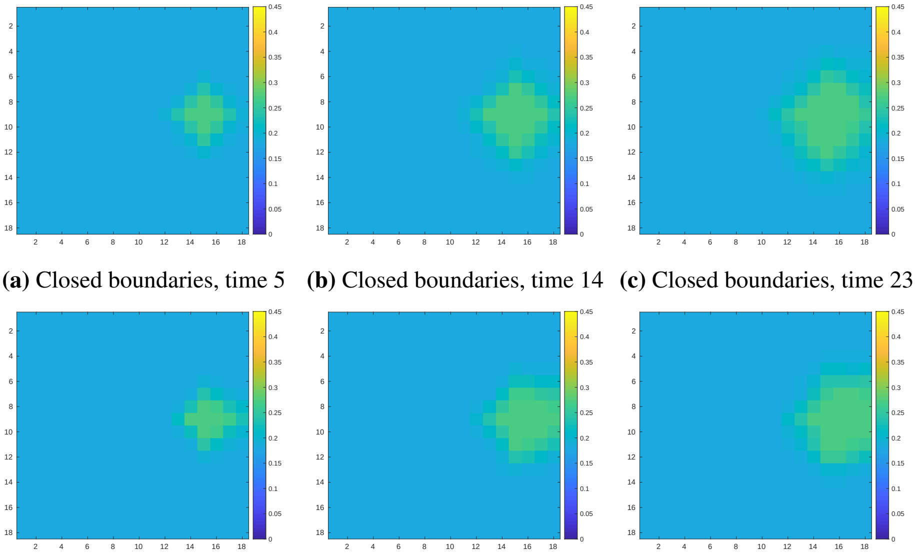
tions at different times. Comparing with figure 4.5, we see that the average of G attributes give a good picture of the average saturations. Figure 4.10 shows G attributes for the same times for one run of simulations for each of the different boundary cases. These simulations runs are the same as in figure 4.6 and 4.8. One is able to see the CO_2 plume in the pictures, but the signals are noisy - more noisy than for the R_0 attributes. This is as expected as the variance is higher for G than for R_0.

4.6 Model training

In this section we build statistical models to use for our estimation of VOI for the seismic data. We will build two kinds of models:

1. KNN models with principal components to estimate values given the seismic data.
2. Neural networks to estimate conditional probabilities of leakage given the seismic data.

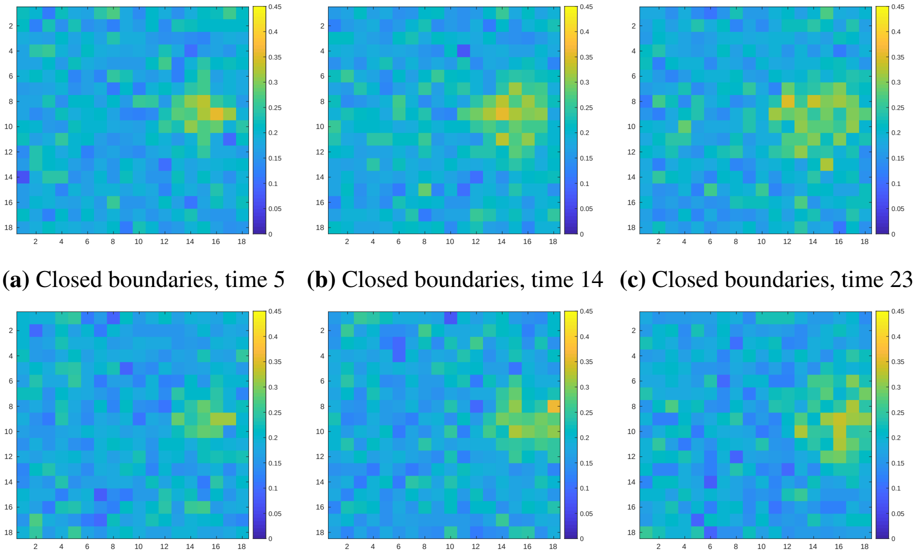
We have extracted data from the simulations for 8 different years during the injection phase. These are years $t = \{2, 5, 8, 11, 14, 17, 20, 23\}$. We want to measure the VOI for seismic surveys done at all these different times, and thus need to train models for all of them. We also want to compare the VOI in the case where we just process the R_0 attribute with the case where both R_0 and G are processed. Thus, we need to fit models for both cases. We call a collection of either R_0 or R_0 and G at a specific time for a data set. We



(a) Closed boundaries, time 5 **(b)** Closed boundaries, time 14 **(c)** Closed boundaries, time 23

(d) Open boundaries, time 5 **(e)** Open boundaries, time 14 **(f)** Open boundaries, time 23

Figure 4.9: Average G attributes for the two different values of x at times 5, 14 and 23.



(a) Closed boundaries, time 5 **(b)** Closed boundaries, time 14 **(c)** Closed boundaries, time 23

(d) Open boundaries, time 5 **(e)** Open boundaries, time 14 **(f)** Open boundaries, time 23

Figure 4.10: G attributes for one simulation run for each value of x at times 5, 14 and 23.

thus have 16 data sets. Each model will be trained on 5000 observations of the simulation data. This set, we call the training set. The remaining data will be used for evaluation and VOI calculations.

The focus in this section is purely on model training. The VOI calculations will be done in the next section.

4.6.1 Value regression with KNN using principal components

For the KNN with PCA modelling, we need to fit two regression models for each data set. One to regress the values when choosing to stop injection of CO_2 ($a = 0$), and one when choosing to continue ($a = 1$).

For each model we need to choose the number of neighbors, k, and the number of principal components (PCs). This we do by comparing the mean square error (MSE) from 5-fold cross-validation on the training data for different numbers of neighbors and PCs. We consider number of PCs in the range 1 to 25, and for each number of PCs we consider k between 1 to 80. For each data set and for each regression, we choose the combination of k and number of PCs giving the lowest MSE.

The values for k and number of PCs for the different models for data set containing only R_0, found through cross-validation, are shown in table 4.1. We notice that for all times, the parameters giving the minimum MSE for the model for $a = 0$ are the same as those for $a = 1$. The values for k and number of PCs for the different models for data set containing R_0 and G, found through cross-validation, are shown in table 4.2. Again, we notice that for all times, the parameters giving the minimum MSE for the model for $a = 0$ are the same as those for $a = 1$.

	$a = 0$			$a = 1$		
Time	k	PCs	MSE	k	PCs	MSE
$t = 2$	58	23	3.561	58	23	188.375
$t = 5$	13	9	8.437	13	9	134.991
$t = 8$	19	13	13.621	19	13	103.656
$t = 11$	14	11	18.567	14	11	82.292
$t = 14$	16	10	25.418	16	10	73.643
$t = 17$	13	11	31.973	13	11	65.251
$t = 20$	13	17	37.974	13	17	57.523
$t = 23$	16	12	43.148	16	12	50.429

Table 4.1: The number of nearest neighbors k and number of principal components chosen for the KNN value regressions with the data sets containing only R_0, for different times and alternatives, along with the corresponding cross-validation MSE.

We now train models on the full sets of training data and use them to predict the values on the 5000 observations taken out for the VOI calculations. We then compute the MSE and MAE for these predictions. The results for the models using only R_0 data are shown in

	$a = 0$			$a = 1$		
Time	k	PCs	MSE	k	PCs	MSE
$t = 2$	79	11	3.690	79	11	195.176
$t = 5$	18	10	7.989	18	10	127.822
$t = 8$	16	7	10.015	16	7	76.213
$t = 11$	13	7	12.830	13	7	56.865
$t = 14$	13	10	19.773	13	10	57.286
$t = 17$	10	11	24.942	10	11	50.903
$t = 20$	11	10	27.827	11	10	42.152
$t = 23$	10	10	32.994	10	10	38.562

Table 4.2: The number of nearest neighbors k and number of principal components chosen for the KNN value regressions with the data sets containing R_0 and G, for different times and alternatives, along with the corresponding cross-validation MSE.

table 4.3. The results for the models using both R_0 and G data are shown in table 4.4. We notice that in general the errors are smaller for the models using both R_0 and G compared to the models using only R_0. We also notice that for models with $a = 1$, the errors generally decrease with time. This is as expected, as for these models the target values do not change with time and we expect that predictions get more accurate as time increases and more CO_2 has been injected into the reservoir. For the models with $a = 0$, we do not see the same trend. This is probably due to the fact that the target values change with time for these models. We see from the value function in section 4.2 that as time increases the difference between the values for leakage and no leakage for $a = 0$ also increases.

	$a = 0$		$a = 1$	
Time	MSE	MAE	MSE	MAE
$t = 2$	3.369	1.573	178.200	11.438
$t = 5$	8.192	2.077	131.065	8.306
$t = 8$	12.428	2.351	94.581	6.485
$t = 11$	17.328	2.407	76.799	5.068
$t = 14$	23.266	2.669	67.406	4.542
$t = 17$	29.825	2.873	60.867	4.104
$t = 20$	35.903	3.013	54.386	3.709
$t = 23$	40.555	3.108	47.398	3.360

Table 4.3: MSE and MAE for predictions on the hold out set for KNN regression models with the data sets containing only R_0, for different times and alternatives.

4.6.2 Classification with CNN

For the binary classification with CNNs, we only need to fit one model for each data set.

	$a = 0$		$a = 1$	
Time	MSE	MAE	MSE	MAE
$t = 2$	3.574	1.626	189.039	11.829
$t = 5$	7.542	1.983	120.670	7.932
$t = 8$	8.623	1.686	65.623	4.651
$t = 11$	12.382	1.743	54.880	3.670
$t = 14$	17.238	2.000	49.943	3.404
$t = 17$	21.869	2.083	44.630	2.976
$t = 20$	25.919	2.162	39.262	2.661
$t = 23$	28.438	2.165	33.237	2.340

Table 4.4: MSE and MAE for predictions on the hold out set for KNN regression models with the data sets containing R_0 and G, for different times and alternatives.

There are many choices to make when fitting a CNN, for instance how many convolutional layers to include and how many filters to have in each. This gives many options of different things to test, when trying to improve the performance of the model. For the current analysis, we will only consider one model architecture for each kind of data, R_0 and (R_0, G). The only parameter we will try to optimize, is the number of epoches to use for training.

For the models using only R_0 data, we will use models set up by the following function:

```
build_model <- function() {
  model <- keras_model_sequential() %>%
    layer_conv_2d(filters = 32, kernel_size = c(3, 3),
        activation = "relu", input_shape = c(18, 18,1)) %>%
    layer_max_pooling_2d(pool_size = c(2, 2)) %>%
    layer_conv_2d(filters = 64, kernel_size = c(3, 3),
        activation = "relu") %>%
    layer_max_pooling_2d(pool_size = c(2, 2)) %>%
    layer_conv_2d(filters = 64, kernel_size = c(3, 3),
        activation = "relu")

  model <- model %>%
    layer_flatten() %>%
    layer_dense(units = 64, activation = "relu") %>%
    layer_dense(units = 1, activation = "sigmoid")

  model %>% compile(
    optimizer = "rmsprop",
    loss = "binary_crossentropy",
    metrics = c("accuracy")
  )
}
```

For the models using both R_0 and G, we will use:

```
build_model2 <- function() {
  model <- keras_model_sequential() %>%
    layer_conv_2d(filters = 32, kernel_size = c(3, 3),
        activation = "relu", input_shape = c(18, 18,2)) %>%
    layer_max_pooling_2d(pool_size = c(2, 2)) %>%
    layer_conv_2d(filters = 64, kernel_size = c(3, 3),
        activation = "relu") %>%
    layer_max_pooling_2d(pool_size = c(2, 2)) %>%
    layer_conv_2d(filters = 64, kernel_size = c(3, 3),
        activation = "relu")

  model <- model %>%
    layer_flatten() %>%
    layer_dense(units = 64, activation = "relu") %>%
    layer_dense(units = 1, activation = "sigmoid")

  model %>% compile(
    optimizer = "rmsprop",
    loss = "binary_crossentropy",
    metrics = c("accuracy")
  )
}
```

Notice that the only difference that we specify, are the size of the input. When using only R_0 data, we pass in to the model a $18 \times 18 \times 1$ tensor. When using both R_0 and G, data we pass in a $18 \times 18 \times 2$ tensor. We here fill one channel with R_0 and the other with G.

In an attempt to find the optimal number of epochs to use for training the model, we do 5-fold cross-validation on the training data. We test numbers of epochs in the range 1-60 and select the numbers giving the lowest value for the cross-validation loss. The results for R_0 data are reported in table 4.5. The results for R_0 and G data are reported in table 4.6. We see from the tables, that for both the models with only R_0 data and for the models with both R_0 and G, the loss, with the optimal numbers of epochs, decreases with time, while the accuracy increases. Accuracy here refers to the proportion of observations sorted into the correct class using a threshold of 50% probability.

We now fit models on the full set of training data and use these to predict the value of x for the data in the hold-out set. The results for models using only R_0 are shown in table 4.7. Generally the loss is decreasing in time, but this decrease is not strictly, as was the case with the cross-validation loss. We also notice that the models are in general better at predicting the no leakage cases than the leakage cases, at least when it comes to assigning them to the correct class. The results for the models using both R_0 and G are shown in table 4.8. We also here notice that the time behaviour of the loss is different than it was for the cross-validation loss and that the models are in general better at assigning the non-leakage cases to the correct class than the leakage cases.

Time	Epochs	Loss	Accuracy
$t = 2$	16	0.504	0.774
$t = 5$	24	0.327	0.865
$t = 8$	23	0.242	0.904
$t = 11$	20	0.187	0.927
$t = 14$	22	0.171	0.936
$t = 17$	20	0.131	0.951
$t = 20$	30	0.119	0.955
$t = 23$	19	0.105	0.957

Table 4.5: The number of epochs chosen for the CNN binary classification with the data sets containing only R_0, for different times, along with the corresponding cross-validation loss and accuracy.

Time	Epochs	Loss	Accuracy
$t = 2$	19	0.502	0.773
$t = 5$	18	0.339	0.856
$t = 8$	20	0.251	0.904
$t = 11$	32	0.189	0.924
$t = 14$	30	0.166	0.933
$t = 17$	35	0.129	0.953
$t = 20$	27	0.111	0.958
$t = 23$	31	0.111	0.960

Table 4.6: The number of epochs chosen for the CNN binary classification with the data sets containing R_0 and G, for different times, along with the corresponding cross-validation loss and accuracy.

Time	Loss	Accuracy	No leakage accuracy	Leakage accuracy
$t = 2$	0.538	0.731	0.778	0.585
$t = 5$	0.328	0.868	0.931	0.670
$t = 8$	0.233	0.901	0.985	0.673
$t = 11$	0.384	0.840	0.806	0.947
$t = 14$	0.388	0.868	1.000	0.456
$t = 17$	0.145	0.944	0.983	0.824
$t = 20$	0.187	0.943	0.996	0.777
$t = 23$	0.091	0.967	0.983	0.919

Table 4.7: Loss, overall accuracy and accuracy for each class of data for predictions on the hold out set for CNN classification models with the data sets containing only R_0, for different times.

The process of fitting a neural network is not a deterministic one. Even with all settings held constant and with the same random seed, two different runs will produce two different models. We illustrate this by comparing the loss on the hold-out set for two different runs for models using only R_0 at different times. This is done in figure 4.11. We see that the losses vary considerably between the runs. On this basis, one could expect high variance in the VOI calculations.

Time	Loss	Accuracy	No leakage accuracy	Leakage accuracy
$t = 2$	0.480	0.784	0.923	0.338
$t = 5$	0.352	0.855	0.896	0.728
$t = 8$	0.249	0.894	0.906	0.857
$t = 11$	0.195	0.932	0.975	0.796
$t = 14$	0.257	0.921	0.994	0.693
$t = 17$	0.196	0.938	0.991	0.771
$t = 20$	0.115	0.957	0.975	0.900
$t = 23$	0.167	0.939	0.932	0.960

Table 4.8: Loss, overall accuracy and accuracy for each class of data for predictions on the hold out set for CNN classification models with the data sets containing R_0 and G, for different times.

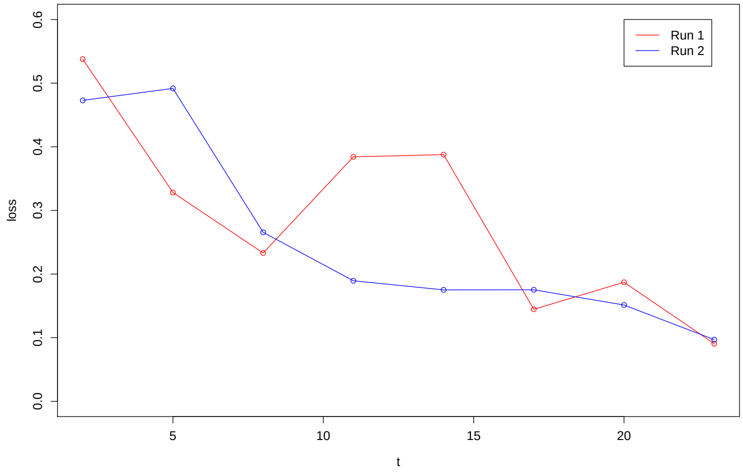


Figure 4.11: Losses from two different runs of the CNN models using only R_0 data.

4.7 Estimating VOI

In this section we estimate the VOI using the approaches presented in section 3.4 with the models trained in section 4.6. We work our way through the different approaches, presenting the VOI calculated for only R_0 and R_0 and G for the 8 different time points. The VOI is calculated using the 5000 observations of data held out from the training process. To get a picture of variability in the estimates, we also train models and calculate the VOI using bootstrap samples of the data. Bootstrap sampling is sampling with replacement from the original data set. We do 100 bootstrap estimates for each time and type of data. For training data, we take bootstrap samples from the original training data and for test data we take bootstrap samples from the original test data.

4.7.1 Approach 1

Figure 4.12 shows the result of the VOI calculations on only R_0 data, both with the original data and with percentiles for the calculations with bootstrap data, using approach 1. We notice that the VOI is estimated to be largest for a seismic survey done at $t = 11$.

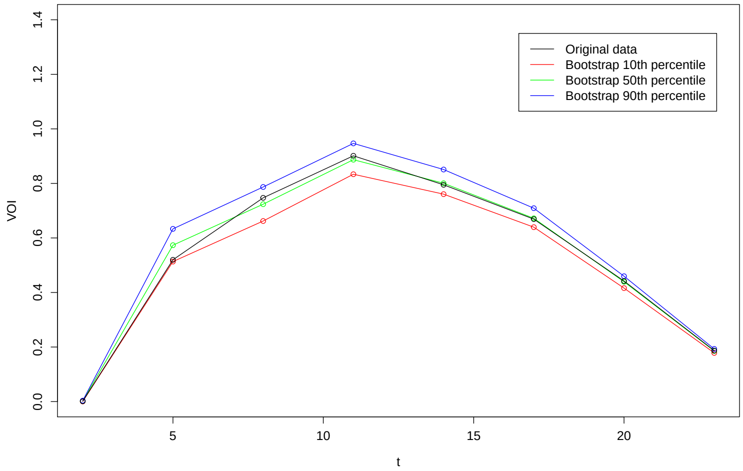


Figure 4.12: VOI calculated using approach 1 for different times with only R_0 data.

Figure 4.13 shows the result of the VOI calculations with R_0 and G data, using approach 1. We have also in this plot included results from the bootstrap estimates. The VOI is estimated to be largest largest when doing the seismic survey at $t = 8$ or $t = 11$.

In figure 4.14, the VOI calculations for both R_0 data and R_0 and G data are plotted. We here compare the 50th percentile from the bootstrap calculations. For most times, the VOI is estimated to be larger for the data sets containing both R_0 and G than the data sets containing only R_0.

Since we do not have analytical results for the VOI at the different times, we are not able to tell how close our results are to the truth. However, our approach have also estimated the prior values, and for these we have analytical results to compare. These comparions are done in figure 4.15 for R_0 data and figure 4.16 for R_0 and G data. For all times, expect $t = 2$ and for both sets of data, the analytical results are not inside the 80% bootstrap interval. Approach 1 has consistently overestimated the prior values.

When doing the bootstrap calculations, training data were sampled from the original training data set, while test data were sampled from the original test data set. Looking at the test data set, we see that there is 1213 observations of leakage out of 5000 observations. This constitutes 24.26% of the observations in the test data set. This means that the prior probability is wrongly represented in the test data set. If the probability of leakage was 24.26%,

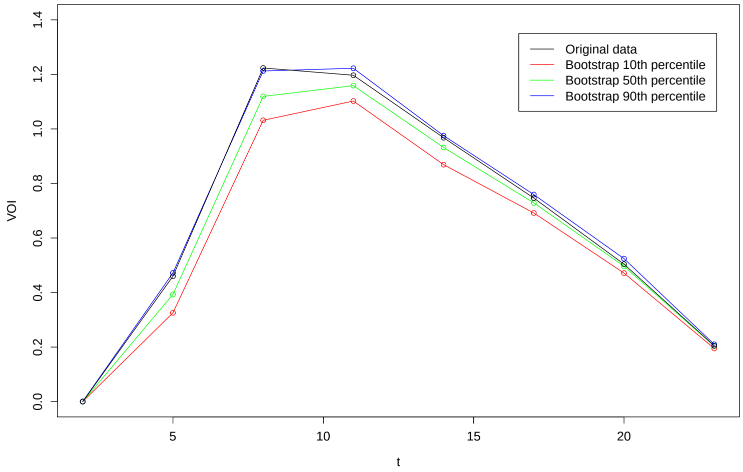


Figure 4.13: VOI calculated using approach 1 for different times with R_0 and G data.

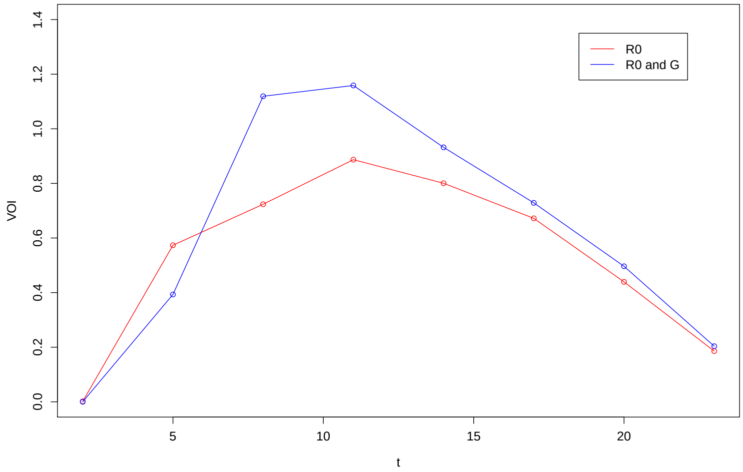


Figure 4.14: 50th percentiles of the bootstrap VOI calculations for different times with R_0 data and R_0 and G data, using approach 1.

the prior value of the decision problem would be -17.76. This is a potential source of bias in the bootstrap estimation of the prior value. However, this unbalance should also affect the posterior value, so in total it is uncertain how the unbalance affects the VOI estimates. We try doing bootstrap sampling again, for models using R_0 data only, this time sampling both training data and test data from the total population. In the total population there is

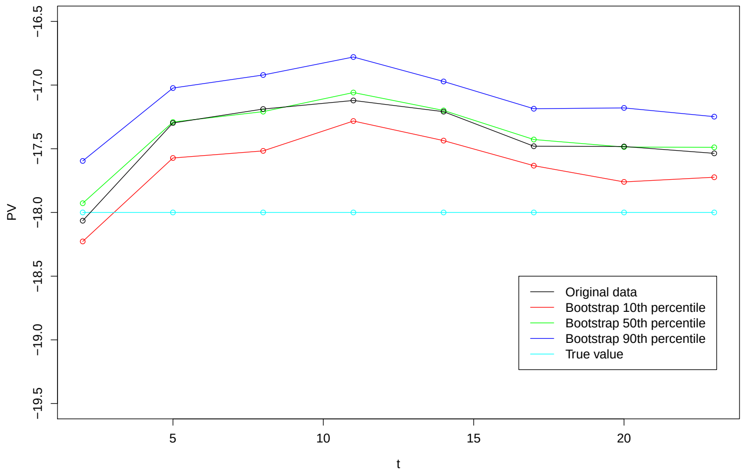


Figure 4.15: Estimation of PV using approach 1 with only R_0 data compared with the analytical results.

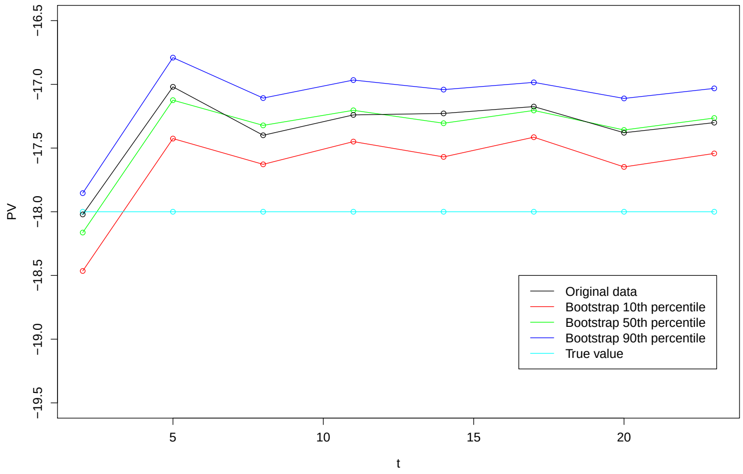


Figure 4.16: Estimation of PV using approach 1 with R_0 and G data compared with the analytical results.

2497 leakages out of 10000 observations, so the prior probability is represented considerably better there. The new bootstrap results for the prior values are shown in figure 4.17, while the new results for the VOI are shown in figure 4.18. In both plots we have included the 50th percentile from the original bootstrap. We see no improvements for the results for the prior value. Also, the VOI estimates do not seem to be affected.

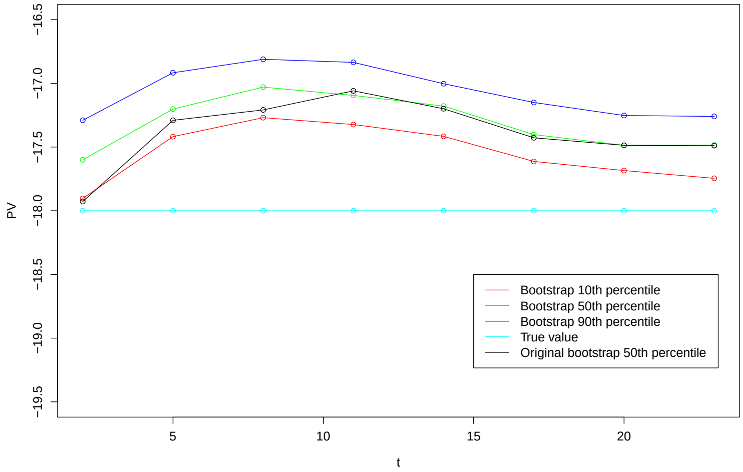


Figure 4.17: Bootstrap results for the prior values, where the bootstrap training data and test data are both sampled from the full data set, for approach 1 with R_0 data.

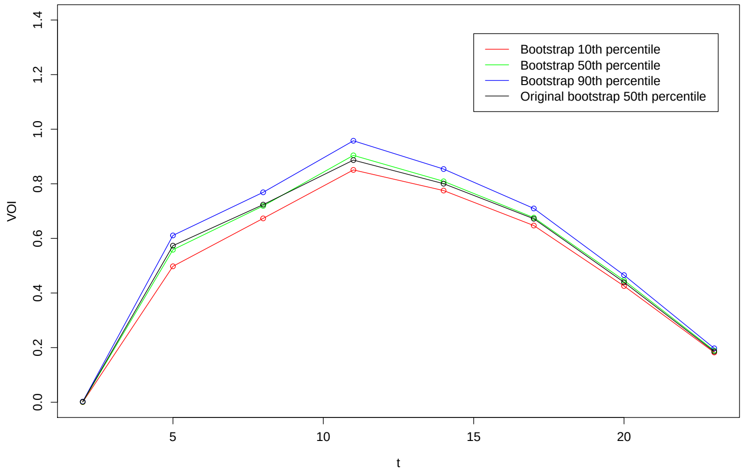


Figure 4.18: Bootstrap results for the VOI, where the bootstrap training data and test data are both sampled from the full data set, for approach 1 with R_0 data.

4.7.2 Approach 2

Figure 4.19 shows the result of the VOI calculations on only R_0 data, both with the original data and with percentiles for the calculations with bootstrap data, using approach 2. We notice large variations in the VOI estimates. The 50th percentile of the bootstrap estimations, reports the VOI to be largest at $t = 5$.

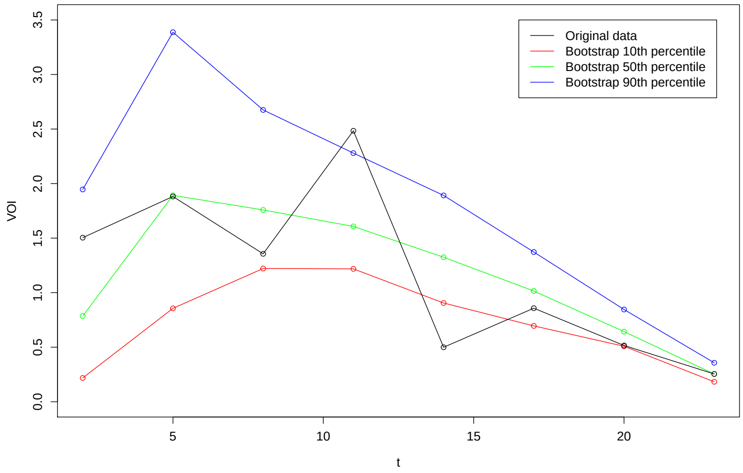


Figure 4.19: VOI calculated using approach 2 for different times with only R_0 data.

Figure 4.20 shows the result of the VOI calculations with R_0 and G data, using approach 2. We have also in this plot included results from the bootstrap estimates. Again we notice large variations in the VOI estimates. Here the 50th percentile of the bootstraps estimations, reports the VOI to be largest at $t = 11$.

In figure 4.21, the VOI calculations for both R_0 data and R_0 and G data are plotted. We here compare the 50th percentile from the bootstrap calculations. From the plot, there is no evidence that the VOI is in general larger for R_0 and G, compared to just R_0.

We now look at the estimates of the prior values for approach 2 compared to the true prior values. This is done for R_0 data in figure 4.22 and R_0 and G data in figure 4.23. We now see that the bootstrap intervals cover the true prior values. However, the variations of the bootstrap estimates are large.

4.7.3 Comparison

In table 4.9 we compare the bootstrap estimates of the VOI between the KNN and CNN approach for data sets containing only R_0 at different times. For most times there is no overlap between the 80% bootstrap intervals for the different approaches. The estimates

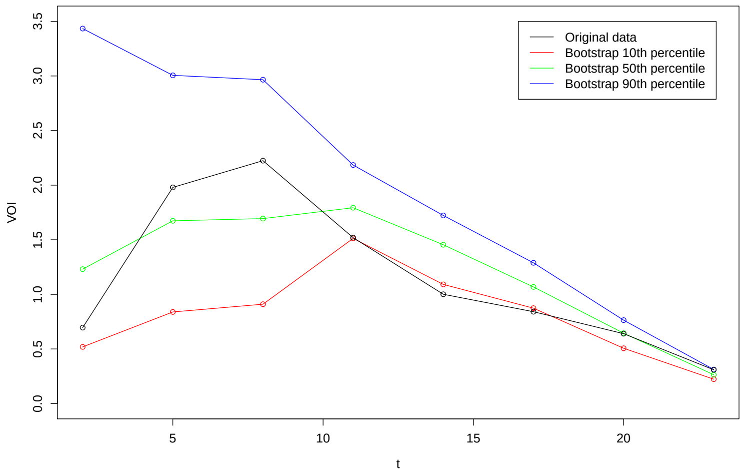


Figure 4.20: VOI calculated using approach 2 for different times with R_0 and G data.

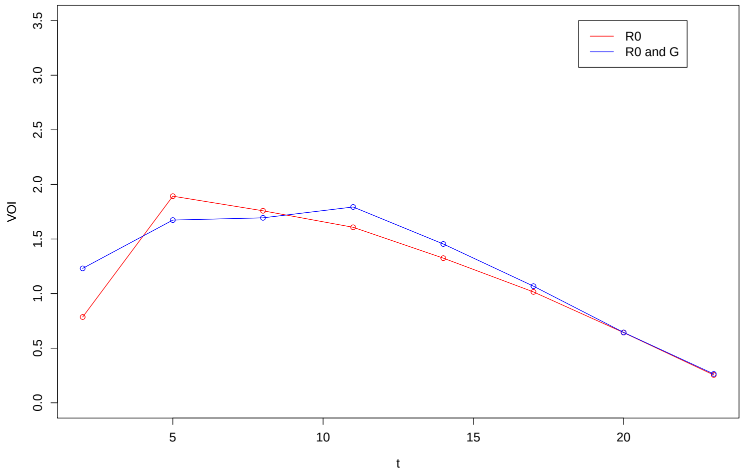


Figure 4.21: 50th percentiles of the bootstrap VOI calculations for different times with R_0 data and R_0 and G data, using approach 2.

using the KNN models are consistently smaller than the estimates using the CNN models. However, what can also be seen, is that the variance in the CNN estimates, are much larger than those of the KNN estimates. In table 4.10 the bootstrap VOI estimates for the different models using both R_0 and G are compared. We see the same trends as with only R_0 data. The bootstrap intervals for the most part do not overlap, with the approach 1

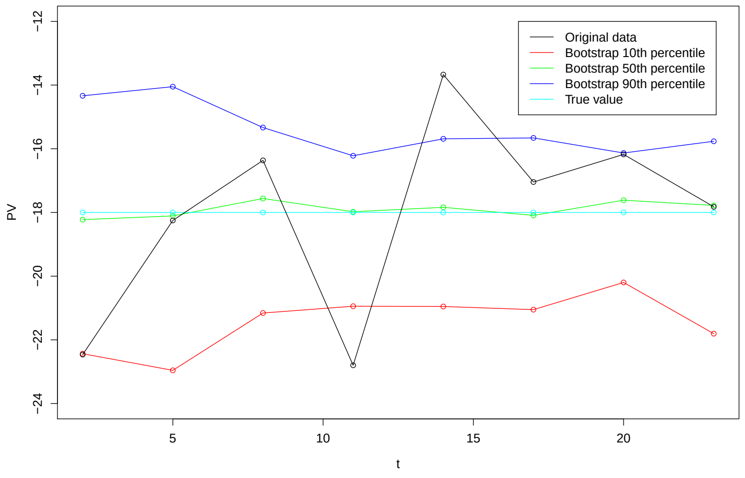


Figure 4.22: Estimation of PV using approach 2 with only R_0 data compared with the analytical results.

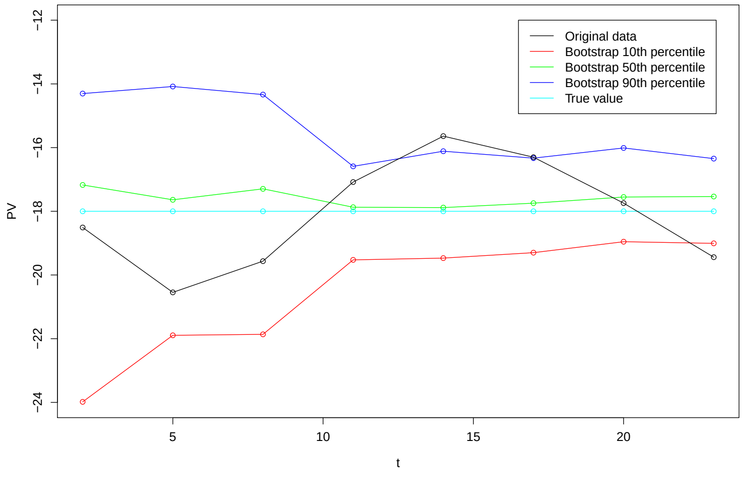


Figure 4.23: Estimation of PV using approach 2 with R_0 and G data compared with the analytical results.

having lower estimates for VOI. Further the variances for approach 2 are large compared to the variances of approach 1.

In figure 4.24 we compare the 50th percentile bootstrap estimate of the two approaches using R_0 data. The value of perfect information has also been included. The consensus

		VOI			
Time	**Model**	10th percentile	50th percentile	90th percentile	Difference
$t = 2$	KNN	0.00	0.00	0.00	0.00
	CNN	0.22	0.79	1.95	1.73
$t = 5$	KNN	0.51	0.57	0.63	0.12
	CNN	0.86	1.89	3.39	2.53
$t = 8$	KNN	0.66	0.72	0.79	0.12
	CNN	1.22	1.76	2.68	1.45
$t = 11$	KNN	0.83	0.89	0.95	0.11
	CNN	1.22	1.61	2.28	1.06
$t = 14$	KNN	0.76	0.80	0.85	0.09
	CNN	0.91	1.32	1.89	0.99
$t = 17$	KNN	0.64	0.67	0.71	0.07
	CNN	0.70	1.02	1.37	0.68
$t = 20$	KNN	0.42	0.44	0.46	0.04
	CNN	0.51	0.64	0.85	0.34
$t = 23$	KNN	0.18	0.19	0.19	0.01
	CNN	0.18	0.25	0.36	0.17

Table 4.9: Bootstrap VOI for the two different approaches at different times using R_0 data. *Difference* refers to the difference between the 90th and 10th percentile.

		VOI			
Time	**Model**	10th percentile	50th percentile	90th percentile	Difference
$t = 2$	KNN	0.00	0.00	0.00	0.00
	CNN	0.52	1.23	3.43	2.92
$t = 5$	KNN	0.33	0.39	0.47	0.15
	CNN	0.84	1.67	3.01	2.17
$t = 8$	KNN	1.03	1.12	1.21	0.18
	CNN	0.91	1.69	2.97	2.06
$t = 11$	KNN	1.10	1.16	1.22	0.12
	CNN	1.51	1.79	2.18	0.67
$t = 14$	KNN	0.87	0.93	0.98	0.11
	CNN	1.09	1.45	1.72	0.63
$t = 17$	KNN	0.69	0.73	0.76	0.07
	CNN	0.87	1.07	1.29	0.42
$t = 20$	KNN	0.47	0.50	0.52	0.05
	CNN	0.51	64	0.76	0.26
$t = 23$	KNN	0.19	0.20	0.21	0.02
	CNN	0.22	0.26	0.31	0.09

Table 4.10: Bootstrap VOI for the two different approaches at different times using R_0 and G data. *Difference* refers to the difference between the 90th and 10th percentile.

between the approaches is that a seismic survey providing only R_0 data should be done after $t = 2$ and before $t = 14$. As time increases, there is a smaller difference between the value of perfect information and the VOI estimates. However, the value of perfect information decreases with time, as more and more CO_2 is put at risk of leakage. In figure 4.25 we do the same comparison for R_0 and G data. The conclusion here would probably be that such a survey has most value at $t = 8$ or $t = 11$. Also here the estimates of VOI get closer to the value of perfect information with time.

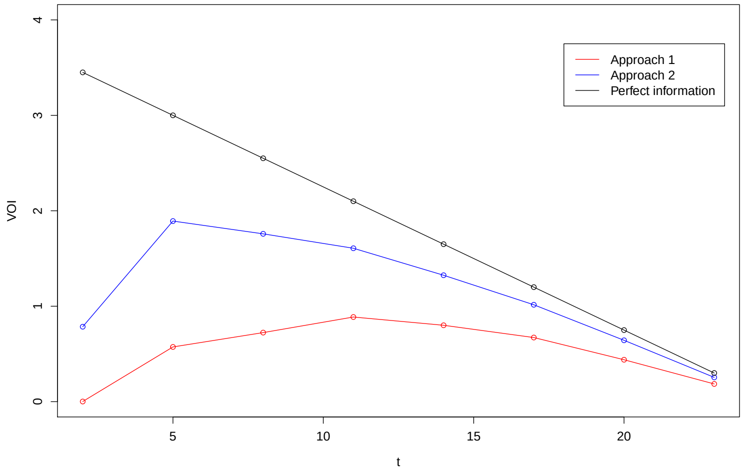


Figure 4.24: 50th percentile bootstrap estimates for each approach for different times using only R_0 data, in addition to the value of perfect information.

4.8 Sensitivity analysis

In this section we test how the noise in the AVO data affects the VOI calculations. We do this by comparing the VOI calculations for R_0 and G data for $t = 11$ using approach 1, for three different values of c. The different values of c are 0.01, 0.04 and 0.16.

We first select the number of principal components and nearest-neighbors to use in the regressions with 5-fold cross-validation as before. The results are shown in table 4.11.

We now train models and calculate the VOI on 100 bootstrap samples for each value of c. The results are shown in figure 4.26. The estimates of VOI decrease with increasing values of c. This is as expected as more noise is added to the AVO data with increasing values of c.

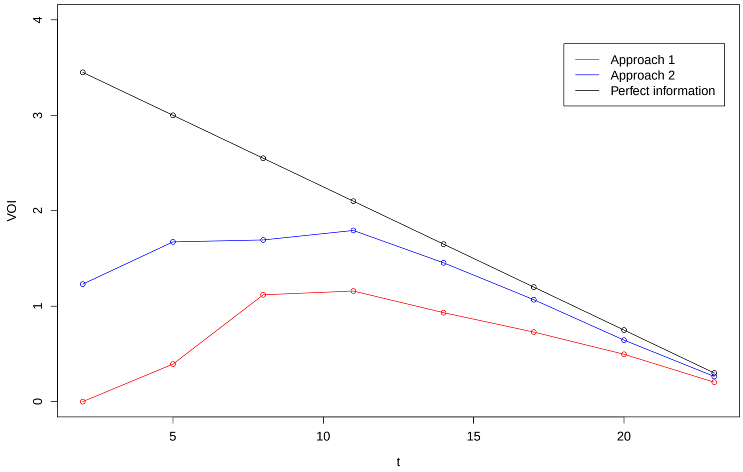


Figure 4.25: 50th percentile bootstrap estimates for each approach for different times using R_0 and G data, in addition to the value of perfect information.

	$a = 0$			$a = 1$		
c	k	PCs	MSE	k	PCs	MSE
0.01	4	15	0.730	4	15	38.592
0.04	13	7	12.830	13	7	56.865
0.16	22	5	2.344	22	5	123.987

Table 4.11: The number of nearest neighbors k and number of principal components chosen for the KNN value regressions with the data sets containing R_0 and G at $t = 11$, for different values of c and alternatives, along with the corresponding cross-validation MSE.

4.9 Summary and discussion

In this chapter we have applied the simulation-regression approach to estimate the VOI for seismic surveys at different times for a constructed case relating to leakage detection of CO_2 at Smeaheia. Reservoir simulations were done in MRST and from simulated saturations, we generated noisy AVO data. Two regression approaches were tested for the value regression, KNN models with principal components of seismic data and CNNs, and for each approach for different times we tested two different data sets of seismic data. These were data sets containing only the R_0 attribute and data sets containing both the R_0 and G attribute. The estimated values from the models were used to estimate prior and posterior values, which again were used to estimate VOI. We also trained models and estimated VOI on bootstrap samples of the data to get an idea of the variability of the VOI estimates.

VOI estimates provided by the KNN models were lower than the ones provided by the

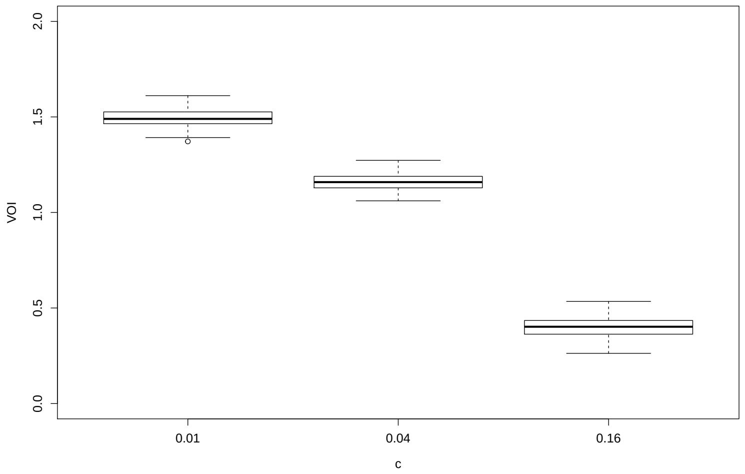


Figure 4.26: Bootstrap estimates of VOI calculated using approach 1 with R_0 and G data at $t = 11$ with varying values of c.

CNN models, and for most cases the bootstrap intervals of VOI estimates did not overlap. It might be that the estimates from one or both of the models were biased. For the prior values we know the truth, and when looking at bootstrap estimates we saw that the KNN models consistently overestimated the prior values. However, an understanding of potential biases in the posterior value calculations would be needed before concluding that the VOI estimates are biased. For the CNN models, we did not see any consistent over- or underestimation of the prior values.

The VOI estimates provided by the KNN models showed quite stable behaviour over the bootstrap samples. This was not the case for the estimates provided by the CNN models. If the VOI were not calculated over bootstrap samples, we would not have been able to get any meaningful results from the CNN approach. However, calculating bootstrap samples as was done here, is computationally expensive.

In theory the VOI should always be higher for data sets containing both R_0 and G, compared to data sets containing only R_0, as VOI should not decrease by added information. This was not always the case for our estimates. From approach 1 it could be concluded that R_0 and G data are in general more valuable than just R_0 data. This is less the case for approach 2.

From the VOI estimates, we are not able to say for sure when a seismic survey should be done, but for both types of data, the general consensus is that a test would have most value some time after $t = 2$ and before $t = 14$. The qualitative behaviour of the VOI curves with time fits well with the nature of the problem. It is reasonable to assume that the value of information will be the highest at some intermediate time point during the injection phase. In the beginning of the injection phase, the value of perfect information will be high, but the test accuracy will be low. At stages towards the end of injection, test accuracy will be high, but the value of perfect information will be low.

Several things could be looked into, in an attempt to optimize the performance of the regression models. Before calculating the PCs, we scaled the seismic data. This might be a wise choice when using both R_0 and G at the same time, as G varies over a larger scale than R_0, but might not be optimal when using just R_0. The scaling was done according to the standard deviations of R_0 and G over the observations in the training data for each cell in the grid. It might be a more optimal choice to scale by the standard deviations for the total population of R_0 and G attributes in the training data respectively. For the CNN models we did not do any scaling of the data, so scaling could also be considered before fitting those models. When passing both R_0 and G data into the neural networks, R_0 and G were treated as different channels. One could also consider merging them into one channel. For the CNN models, we only considered one instance of architecture when it came to things such as number of different kinds of layers and units. A thorough analysis could be done, to optimize the architecture. However, this would be quite time consuming for the given case, considering all the different models that were fitted and all the different things that could be changed in the model architecture. In general, 5000 observations is a quite low number when fitting CNNs and this might be a cause of the large variability that we see in the VOI estimates from these models. Further, what could be a problem for both techniques, is the unbalance of leakages compared to non-leakages in our data. Further investigation could be done, to see how this affects the performance of the models.

Chapter 5

Conclusion

In this work we have done VOI calculations of seismic data in the context of CO_2 storage and detection of potential CO_2 leakage. A case was developed where a decision maker must decide when in time to do a seismic survey, and based on the results of the survey decide between continuing or stopping the injection of CO_2. Accuracy of a seismic survey, is likely to increase with time and amount of CO_2 injected into the reservoir. On the other hand, the cost of leakage is likely to increase with increasing amounts of CO_2 injected into the reservoir. In this sense, the decision of when to do the survey becomes a trade-off between test accuracy and amount of CO_2 put at risk of leakage. For the case study we have used Smeaheia, which is located in the North-Sea and is a site being considered as storage location for the Norwegian full-scale CCS project.

To estimate the VOI for seismic surveys at different times, we have used the simulation-regression approach. 10000 simulations of CO_2 injection and storage in Smeahiea was done using MRST. Two different cases were simulated - open boundary cases, which lead to leakage, and closed boundary cases which do not lead to leakage. In addition porosities and permeabilities were randomly permuted between the simulations. Based on CO_2 saturations from the reservoir simulations, noisy seismic AVO data were generated. The next step was to regress values on the seismic data, in order to estimate posterior and prior values for the VOI calculations. Two regression techniques were tested - KNN models using principal components of the data and CNNs.

The VOI estimates using KNN models were low compared to the estimates using CNNs. On the other hand, the estimates from the CNNs had much higher variance. Comparing the VOI results using the KNN models with the ones using the CNNs, it is not unreasonable to suspect that the KNN results are downwards biased. For further work, other statistical methods could be considered to estimate the VOI in this case. One could also try doing the calculations with a larger data set to see if this decreases the variability of the VOI estimates from the CNN models.

In conclusion our VOI estimates did not give us a precise answer of when a seismic survey has the most value, but we could see that a seismic survey should not be done to early or to late in the injection process. This is also consistent with the time dependent VOI example presented in the theory part.

We have demonstrated a fairly general workflow, but inputs like for instance costs in the decision problem and the prior probability would need to be adjusted on a case to case basis. These are inputs that will influence the VOI.

A natural next step is to develop a case and a workflow where sequential surveys are considered. In real world scenarios it is probable that one will consider doing more than one seismic survey during the injection phase. This would involve the prior probability being updated when a survey has been done, and would be a more complex case than what has been considered here.

Bibliography

Allaire, J., Chollet, F., 2019. keras: R Interface to 'Keras'. R package version 2.2.4.1. URL `https://CRAN.R-project.org/package=keras`

Arts, R., Eiken, O., Chadwick, A., Zweigel, P., Van der Meer, L., Zinszner, B., 2004. Monitoring of co2 injected at sleipner using time-lapse seismic data. Energy 29 (9-10), 1383–1392.

Avseth, P., Mukerji, T., Mavko, G., 2005. Quantitative seismic interpretation: Applying rock physics tools to reduce interpretation risk. Cambridge university press.

Brie, A., Pampuri, F., Marsala, A., Meazza, O., et al., 1995. Shear sonic interpretation in gas-bearing sands. In: SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers.

Chollet, F., Allaire, J. J., 2018. Deep Learning with R, 1st Edition. Manning Publications Co., Greenwich, CT, USA.

Dupuy, B., Ghaderi, A., Querendez, E., Mezyk, M., et al., 2017. Constrained avo for co2 storage monitoring at sleipner. Energy Procedia 114, 3927–3936.

Dupuy, B., Querendez, E., Ghaderi, A., Romdhane, A., Eliasson, P., 2018. Norwegian large-scale co2 storage project (smeaheia): Baseline geophysical models. In: 14th Greenhouse Gas Control Technologies Conference Melbourne. pp. 21–26.

Dutta, G., Mukerji, T., Eidsvik, J., 2019. Value of information analysis for subsurface energy resources applications. Applied Energy 252, 113436.

Eidsvik, J., Dutta, G., Mukerji, T., Bhattacharjya, D., 2017. Simulation–regression approximations for value of information analysis of geophysical data. Mathematical Geosciences 49 (4), 467–491.

Eidsvik, J., Martinelli, G., Bhattacharjya, D., 2018. Sequential information gathering schemes for spatial risk and decision analysis applications. Stochastic environmental research and risk assessment 32 (4), 1163–1177.

Eidsvik, J., Mukerji, T., Bhattacharjya, D., 2015. Value of information in the earth sciences: Integrating spatial modeling and decision analysis. Cambridge University Press.

Furre, A.-K., Eiken, O., Alnes, H., Vevatne, J. N., Kiær, A. F., 2017. 20 years of monitoring co2-injection at sleipner. Energy procedia 114, 3916–3926.

Howard, R. A., 1966. Information value theory. IEEE Transactions on systems science and cybernetics 2 (1), 22–26.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An introduction to statistical learning. Vol. 112. Springer.

Keeling, C. D., Piper, S. C., Bacastow, R. B., Wahlen, M., Whorf, T. P., Heimann, M., Meijer, H. A., 2001. Exchanges of atmospheric co2 and 13co2 with the terrestrial biosphere and oceans from 1978 to 2000. i. global aspects.

Lie, K.-A., 2019. An introduction to reservoir simulation using MATLAB/GNU Octave: User guide for the MATLAB Reservoir Simulation Toolbox (MRST). Cambridge University Press.

Nordbotten, J. M., Celia, M. A., 2011. Geological storage of CO2: modeling approaches for large-scale simulation. John Wiley & Sons.

Norwegian Petroleum Directorate, 2014. CO2 storage atlas. https://www.npd.no/en/facts/publications/co2-atlases/co2-atlas-for-the-norwegian-continental-shelf/, [Online; accessed 1-July-2019].

Sato, K., 2011. Value of information analysis for adequate monitoring of carbon dioxide storage in geological reservoirs under uncertainty. International Journal of Greenhouse Gas Control 5 (5), 1294–1302.

SINTEF, 2016a. Fully-implicit VE simulation. https://www.sintef.no/projectweb/mrst/modules/co2lab/ve-models/fully-implicit-ve-simulation/, [Online; accessed 1-July-2019].

SINTEF, 2016b. MRST co2lab, Modeling basics. https://www.sintef.no/projectweb/mrst/modules/co2lab/modeling-basics/, [Online; accessed 1-July-2019].

Strong, M., Oakley, J. E., Brennan, A., 2014. Estimating multiparameter partial expected value of perfect information from a probabilistic sensitivity analysis sample: a nonparametric regression approach. Medical Decision Making 34 (3), 311–326.

United Nations, 2019. Global Issues Overwiev. https://www.un.org/en/sections/issues-depth/global-issues-overview/index.html, [Online; accessed 1-July-2019].