BACKCALCULATION OF INTELLIGENT COMPACTION DATA FOR THE MECHANICAL PROPERTIES OF SOIL GEOSYSTEMS

AFSHIN GHOLAMY SALEHABADY

Doctoral Program in Geological Sciences

APPROVED:

Laura Serpa, Ph.D., Co-Chair

Vladik Kreinovich, Ph.D., Co-Chair

Aaron Velasco, Ph.D.

Hector Gonzalez, Ph.D.

Soheil Nazarian, Ph.D.

To my Mother, Father, and to my brothers:

Maziar, Arash, Behzad, and last but not least, Behrouz.

This humble work is sign of love to you.

And to my scholar advisers,

Laura Serpa and Vladik Kreinovich.

BACKCALCULATION OF INTELLIGENT COMPACTION DATA FOR THE MECHANICAL PROPERTIES OF SOIL GEOSYSTEMS

by

AFSHIN GHOLAMY SALEHABADY, M.Sc. in Geological Sciences

DISSERTATION

Presented to the Dissertation Committee

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Doctoral Program in Geological Sciences

THE UNIVERSITY OF TEXAS AT EL PASO

December 2018

Acknowledgements

First, and foremost, I would like to express my profound gratitude to my mentor Professor Laura Serpa. I thank you for all your support, guidance, and friendship over the last two years. You took a chance on me during a time in my life when I was anything but a sure bet. For that, I am forever in your debt. I also wish to thank my Committee Co-Chair, Dr. Vladik Kreinovich from Computer Science Department, and Dr. Soheil Nazarian from Civil Engineering Department, for their support and encouragement. Many thanks to Dr. Hector Gonzalez and Dr. Aaron Velasco for their insightful comments.

I would like to thank all the faculty, staff, and students of the Departments of Geological Sciences and Civil Engineering at The University of Texas at El Paso for all their hard work, dedication, and help.

Last but not least, I would like to thank my family – my mother and my brothers – for the endless support throughout my life.

NOTE: This dissertation was submitted to my Supervising Committee on November X, 2018.

Abstract

For national economy, it is very important to have a reliable infrastructure. Because of this, all over the world, new roads are constantly being built and old roads are being maintained and, if necessary, expanded and/or repaired. Building a good quality road is very expensive, it costs several million dollars per kilometer. It is therefore crucial to make sure that the newly built and newly repaired roads are sufficiently stiff – so that they can withstand the predicted volume of traffic for a sufficient number of years.

Current methods of estimating the stiffness are time-consuming and labor-consuming. The most accurate technique is to take a sample from the compacted subgrade or base, bring it to the lab, and measure the mechanical parameters that characterize the corresponding stiffness – this takes days. Another possibility is to measure the road stiffness on-site. There are several different measuring techniques for such measurements, but they are all very labor-intensive and often take days to acquire and process the data. The main idea of *intelligent compaction* is to measure the road's mechanical properties while the road is being compacted, by placing accelerometers on the rollers and/or geophones (sensors for detecting ground movement) at different depths at several locations.

The main challenge that prevents intelligent compaction from being a widely accepted road building technique is that the relation between the mechanical properties of the soil and the resulting accelerations is very complex, it is described by a system of dynamic non-linear partial differential equations. It is therefore desirable to determine the desired characteristic in real time, without the need to solve the corresponding system of partial differential equations. This is the main task that we perform in this study. In the process of implementing this task, we also solve several auxiliary tasks which can be used in a more general data processing setting.

Table of Contents

Chapter 1

Formulation of the Problem

1.1 Need to Determine Mechanical Properties of Earthworks During Road Construction: Formulation of a Practical Problem

For national economy, it is very important to have a reliable infrastructure. Because of this, all over the world, new roads are constantly being built and old roads are being maintained and, if necessary, expanded and/or repaired. Building a good quality road is very expensive, it costs several million dollars per kilometer. It is therefore crucial to make sure that the newly built and newly repaired roads can withstand the predicted volume of traffic for a sufficient number of years.

Most roads consist of several layers. The road is built on top of the soil. Soil is rarely stiff enough, so usually, the soil is first compacted – and, if needed, additional stiffening materials called *stabilizers* (or *treatments*) are added to the soil before compaction. The resulting layer of compacted soil is known as the *subgrade.*

On top of the subgrade, additional stiff material is placed, called a *base*, it is usually gravel. The base is also compacted, to make it even stiffer.

The base is usually reasonably thick – between 15 to 30 cm. It is difficult to compact a layer of such thickness, so usually, instead of placing all the base at once, practitioners place first a thinner layer of the base material, compact it, then place another thinner layer, etc., until they reach the desired thickness.

Finally, asphalt or concrete is placed on top of the base.

For the road to be of high quality, all three layers must be sufficiently stiff. For example, if the subgrade is not yet stiff enough, we should compact it further – or add more stabilizer – to make it sufficiently stiff, otherwise the resulting road will not be of good quality.

Current methods of estimating the stiffness are time-consuming and labor-consuming. The most accurate technique is to take a sample from the compacted subgrade or base, bring it to the lab, and measure the mechanical parameters that characterize the corresponding stiffness. Since most roads are built in areas which are far from the nearby labs, this procedure usually takes days. While the road is being tested, the road building company can either keep the road building equipment idle – which will cost money – or move it to a new location, in which case there is a risk that, based on the lab results, there will be a need to move the equipment back to do some more compaction, which also costs a lot of money. To minimize this risk, companies usually over-compact the road – which also leads to additional costs. And it is important to realize that most roads are built by contractors paid by the taxpayer's money, so additional unnecessary costs in road construction are additional costs to taxpayers, costs that could be used for other useful purposes.

Another possibility is to measure the road stiffness on-site. There are several different measuring techniques for such measurements, including light-weight deflectometers (LWD; see, e.g., [45], falling-weight deflectometers (FWD; see, e.g., [42, 65]), dynamic cone penetrometers (DCP), neutron density gauges (NDG), etc.; see, e.g., [48]. All these techniques are very labor-intensive, and while they are somewhat faster than bringing the sample to the lab, they also take days to acquire and process the data. Besides, in contrast to the lab measurements, these techniques do not directly measure stiffness, they measure density and other parameters based on which we can only make very approximate estimates of the desired road stiffness.

In addition, all the existing methods – both lab-based and on-site – are spot tests, they only gauge the road stiffness at certain points. Thus, if the road has a relatively small weak spot, these methods may not detect it – and based on these methods, we may erroneously certify this road as ready for exploitation. Such a faulty road may soon require costly

maintenance – again, at the taxpayers' expense.

1.2 Intelligent Compaction: Main Idea

The main idea of *intelligent compaction* (see, e.g., [3, 14, 49, 52, 53]) is that we can measure the road's mechanical properties while the road is being compacted, by placing accelerometers on the rollers and/or geophones (sensors for detecting ground movement) at different depths at several locations. Based on the results of the corresponding measurements, we can, in principle, determine the mechanical properties of the road at all the locations.

1.3 Intelligent Compaction: Challenges

The main challenge that prevents intelligent compaction from being a widely accepted road building technique is that the relation between the mechanical properties of the soil and the resulting accelerations is very complex, it is described by a system of dynamic non-linear partial differential equations. Even in an ideal situation, when we know all the mechanical characteristics of the subgrade and of the base, it takes several hours on an up-to-date computer to find the corresponding accelerations, and what we want is even more complex: we want to perform *back-calculation*, to solve the *inverse* problem of determining the mechanical characteristics based on the corresponding accelerations; this will take even longer.

1.4 What We Plan to Do

What we need is a way to determine the desired characteristic in real time, without the need to solve the corresponding system of partial differential equations.

1.5 The Resulting Tasks: A Brief Description

In line with the above formulation of the problem, we need to contribute to the solution of the two main problems:

- first, for the 1-layer (subgrade) case, we need to determine the corresponding characteristics of stiffness based on the acceleration measurements;
- second, for the 2-layer (subgrade + base) case, once we have started compacting the base, we need to determine the mechanical characteristics of the base layer based on the measured acceleration (and on the already-determined characteristics of the subgrade).

Let us explain, in more detail, what is needed for these tasks.

1.6 Towards a Detailed Description of the Tasks

First, let us discuss what exactly mechanical characteristics we need.

At first glance, this question may seem easy since a similar problem has been actively studied in civil engineering in general and, in particular, in the analysis of earthworks related to construction of buildings, bridges, dams, etc. However, from the mechanical viewpoint, road-related problems are different. For example, in building construction, we have a reasonably constant stress on the underlying soil, while for a road, we have a fast changing stress when a truck goes over this section of the road at a reasonably high speed. To capture the effect of such dynamic loads on different constructions, we need to find the value of the corresponding elastic modulus E. In situations when this modulus is measured in the lab, it is called the *resilient modulus* E.

From this viewpoint, our objective is to estimate the values of the elastic modulus in both layers at different locations.

One of the challenges here is that the usual partial differential equations that describes the reaction of a media to different forces use different mechanical characteristics. It is

therefore important to understand how the elastic modulus depends on the more traditional mechanical characteristics of the medium. There exists several models for such dependence. At present, the empirical comparison between different models seems to indicate that one of these models is the most adequate:

$$E = k_1' \cdot \left(\frac{\theta}{P_a} + 1\right)^{k_2'} \cdot \left(\frac{\tau_{\text{oct}}}{P_a} + 1\right)^{k_3'},$$

where

$$\theta = \sigma_1 + \sigma_2 + \sigma_3$$

is the *bulk stress*,

$$\tau_{\text{oct}} = \frac{1}{3} \cdot \sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_1 - \sigma_3)^2 + (\sigma_2 - \sigma_3)^2}$$

is the *octahedral shear stress*, and σ_1, σ_2, and σ_3 are principal stresses. This model was proposed in Ooi et al. [57], and in Mazari et al. [44], it was shown to be the most adequate model for describing the elastic modulus.

Usually, the parameters k_2' and k_3' are determined by the material – e.g., whether it is clay or different type of gravel – while the parameter k_1' varies strongly even for the same material – e.g., for gravel, the value of k_1' depends on how big the grains are, what is their density, etc. Thus, once we know the substance forming the soil and/or material used for the base layer, we thus know the corresponding values k_2' and k_3' – but not the corresponding values of k_1'.

We want to determine the elastic modulus based on the measured values of acceleration. To enhance compaction, the roller vibrates with a frequency between 20 and 60 Hz. So, the whole process is periodic with this frequency, and the measured acceleration is also periodic with the same frequency. So, to eliminate the noise, it is reasonable to perform a Fourier transform, and to keep only the components corresponding to this known frequency. The resulting information can be equivalently described in terms of the displacement.

From this viewpoint, we face the following two tasks:

Task 1: *for the 1-layer case, determine the elastic modulus E based on the displacement d_1 and on the mechanical characteristics k'_2 and k'_3 of the subgrade.*

Task 2: *for the 2-layer case, determine the elastic modulus E of the base layer by using the 2-layer displacement d_2, the previously measured 1-layer displacement d_1, parameters k'_{2s} and k'_{3s} of the subgrade, and parameters k'_{2b} and k'_{3b} of the base layer.*

We need fast techniques for solving these two problems. So, we need simple expressions for the corresponding solutions. There are two ways of getting such expressions:

- a traditional idea is to use the corresponding physics to come up with possible terms, and then use regression techniques to find the parameters that best fit the empirical data;
- an emerging approach of machine learning – which now means mostly neural networks (see, e.g., [2, 29, 46, 47]) – is not to fix any specific terms, but rather let the computer find the terms which are empirically most appropriate.

In our research, we use both approaches, and select the best of the resulting models.

These tasks are particular cases of the general *back-calculation problem*; see, e.g., [30, 34, 35, 47, 50, 51, 72, 73].

1.7 Auxiliary Tasks

The first auxiliary task is related to the fact that while empirically, this model is known to be the best, to be on the safe side, it is desirable to have some theoretical justification for this formula – to make sure that this formula is indeed in line with the general theory and that we are not missing any possible more accurate expression. Coming up with such a theoretical explanation is our first task.

Task 3: *find a theoretical justification for the empirical formula describing the dependence of elastic modulus on other mechanical characteristics.*

The second auxiliary task pertains to the fact that our formulas that solve the main Tasks 1 and 2 are based mostly on the results of computer simulations. Computer simulations are inevitably simplifying: for example, such models usually assume that each layer is homogeneous, while in reality, the properties of each layer somewhat change from one location to another – especially when we are talking about the subgrade, which is often nothing else but the compressed original soil. It is therefore reasonable to compare the simulation results with the actual measurement results. Such a comparison was performed in [44] for a similar problem – of analyzing the Light Weight Deflectometer (LWD) measurements. It turned out that the simulation results overestimate the stiffness: often, by a factor of 4. So, for that problem, to provide guaranteed bounds, we need to divide the simulated results by a factor of 4 – in other words, to use the safety factor of 4. Since our problem is similar, we expect that the same safety factor should work in our case as well – and our preliminary results show that this is indeed the case. However, to be on the safe side, it is desirable to have a theoretical justification for this empirical safety factor. So, we arrive at:

Task 4. *provide a theoretical justification for the empirical safety factor.*

Other auxiliary tasks go beyond this particular problem, towards more general settings. In general, how can we find simple formulas that best describe the results of experiments and simulations?

- First, we perform measurements and/or run the corresponding simulations.
- After that, according to the usual statistical approaches, we divide the results of measurement and simulation into the training set and the testing set (and the validation set).
- Then, we select a criterion that formalizes what we mean by "best".
- After that, we use the training set and the selected criterion to come up with appropriate formulas; these formulas are then checked on the testing and validation

sets.

- Finally, if needed, we visualize the results, to make them clearer to the users.

In all these steps, we face important auxiliary tasks.

First, we need to determine how many simulations to run. This is an important issue: if we run too few simulations, we may not reach the desired accuracy, but if we run too many simulations, we will be wasting computation time that could be used, e.g., to run more accurate simulation models.

Task 5: *determine the appropriate number of simulations.*

Then, we need to decide on the best way to divide the sample into training, testing, and validation sets. This is also a very important issue: if we select too many values for the testing and validation sets, we leave too few value for the training set and, as a result, the models obtained by analyzing this set will not be very accurate. On the other hand, if we keep too few values in the testing and validation sets, we will have too few samples to get an adequate validation of the resulting models.

Task 6: *come up with the most adequate division into training, testing, and validation sets.*

Once we have measurement and simulation results, and we have selected a training set from these results, we need to come up with a model that most adequately describes these results. For this purpose, we need to select a criterion for checking how adequate is each model.

The existing back-calculating techniques have been designed for problems for which the range of the corresponding values is reasonably small. In such cases, to gauge how accurate the model is, it is reasonable to simply take the difference between the actual and predicted values. Because of this, most existing packages use the mean squared value of such a difference to find the appropriate model.

In pavement engineering, the elastic modulus can change by orders of magnitude – especially when we take into account that the subgrade can differ from an already stiff

rocky soil to a very soft clay. In such situations, minimizing the mean square difference of the absolute accuracy does not make too much sense – large values corresponding to stiff materials will dominate, and the small differences corresponding to an important case of soft subgrade will be ignored. In such situations, it is more appropriate to use *relative* errors, i.e., approximation errors described in terms of percentages of the original values.

In principle, we could re-write the existing software packages so that they take into account relative error – but this is time-consuming, and besides, some of these packages are proprietary, they do not provide the users with the codes that we could modify. It is therefore desirable to come up with ways of using the absolute-error minimization techniques to solve relation-error minimization problems. This is our first auxiliary task:

Task 7: *come up with ways of using the absolute-error minimization techniques to solve relative-error minimization problems.*

Once the sample is selected and the criterion is selected, we need to actually find the model that most adequately fits the selected data. As we have mentioned, in our analysis, we use both the traditional statistical methods and the soft computing methods such as neural networks.

Traditional statistical methods have been used for decades and centuries. With these methods, it is usually clear which techniques work best. In contrast, neural techniques are still being developed; it is not always clear which techniques are the best. Thus, we arrive at yet another auxiliary task:

Task 8: *come up with neural network techniques which are most adequate for our data processing problems.*

Finally, once the results are ready, it is desirable to visualize them. Indeed, most of the techniques provide numbers. For engineers and practitioners to be able to use and understand these numbers, it is desirable to *visualize* them. In particular, since most of the estimates and predictions are probabilistic in nature, it is reasonable to be able to plot the corresponding histogram, to give the user a clear understanding of the corresponding prob-

ability distribution. Here, we face another challenge: to come up with a clear histogram, we need to select the appropriate bin size. If the bin size is too small, the resulting histogram is chaotic, and does not give us a good understanding of the corresponding probability distribution. If this bin size is too large, we get a good general picture, but we may miss important details. In probability and statistics, there are methods of selecting optimal bin sizes, but these methods assume that we already have a lot of information about the probability distribution – and in our problem, like in many other engineering tasks, we do not have this information. It is therefore important to solve the following task:

Task 9: *come up with a general technique for selecting the optimal bin size for a histogram, a technique that does not require that we have any prior information about the corresponding probability distribution.*

1.8 Tasks: Summary

In our research, we have the following five tasks:

- **Task 1:** *for the 1-layer case, determine the elastic modulus E based on the displacement d_1 and on the mechanical characteristics k'_2 and k'_3 of the subgrade.*
- **Task 2:** *for the 2-layer case, determine the elastic modulus E of the base layer by using the 2-layer displacement d_2, the previously measured 1-layer displacement d_1, parameters k'_{2s} and k'_{3s} of the subgrade, and parameters k'_{2b} and k'_{3b} of the base layer.*
- **Task 3:** *find a theoretical justification for the empirical formula describing the dependence of elastic modulus on other mechanical characteristics.*
- **Task 4.** *provide a theoretical justification for the empirical safety factor.*
- **Task 5:** *determine the appropriate number of simulations.*
- **Task 6:** *come up with the most adequate division into training, testing, and validation sets.*

- **Task 7:** *come up with ways of using the absolute-error minimization techniques to solve relative-error minimization problems.*
- **Task 8:** *come up with neural network techniques which are most adequate for our data processing problems.*
- **Task 9:** *come up with a general technique for selecting the optimal bin size for a histogram, a technique that does not require that we have any prior information about the corresponding probability distribution.*

It should be mentioned that, in contrast to Tasks 1–3 which are specific for pavement engineering applications, Tasks 4–8 are of more general interest, their results can be used in many other problems of science and engineering.

1.9 Data

To work on these tasks, we use both the actual measurement results and the results of computer simulation. In our opinion, while the actual measurement results are important – to provide the ultimate tests of how good our results are – the most important part of the data comes from simulations. This is different from many typical geophysical problems, where models are very approximate: in our models, we know exactly what materials are at each depth, and we use the equations of mechanics, which are known to be very accurate in describing mechanical properties – while in geophysics, we often do not have a good understanding of what material is at different depths, what is the shape of the borderline between different materials, etc.

1.10 Structure of the Dissertation

In this dissertation, we present solutions to all the tasks. Solutions to Tasks 1 and 2 are presented in Chapter 2, solution to Task 3 through 9 are presented, correspondingly, in

Chapters 3 to 9.

Chapter 2

Inverse Problem for Intelligent Compaction: Main Results

2.1 Inverse Problem for Intelligent Compaction: Reminder

The main objective of this research is to develop methods for evaluating the quality of the pavement in a timely manner.

We are considering a typical case of a pavement consisting of two layers:

- the subgrade and
- the base (placed over this subgrade).

In the process of road construction of a 2-layer pavement:

- First, the subgrade is reinforced (if needed) and then compacted.
- Then, the base is placed on top of the subgrade, and the pavement is compacted again.

On each of these two compaction stages, sensors (accelerometers) are placed on the rollers. The accelerations measured by these sensors are then used to gauge the quality of the pavement.

A good quality pavement should have:

- a sufficiently stiff subgrade and

- a sufficiently stiff base.

Both for the subgrade and for the base, it is important to make sure:

- that in all the spatial locations, the pavement is stiff enough, and
- also that the stiffness is uniform across the pavement – otherwise, the traffic load will be unequally distributed, leading to too much stress (and earlier wear) for some locations.

For the *subgrade*, its stiffness can be extracted directly from the "pre-mapping", i.e., from the measurements performed while the subgrade is being compacted. Specifically, from on the sensors attached to the compacting roller, we can determine the deflection d_1 caused by the stress. The stiffness of the subgrade at each spatial location can then be determined by dividing the known force of the roller by the deflection at this particular location.

In contrast, the stiffness of the *base* cannot be determined directly from the measurements. From the sensors attached to the roller that compacts the 2-layer pavement, we can extract the deflection d_2 that describes the stiffness of the 2-layer pavement. For each spatial location, we need to evaluate the stiffness of the base from:

- the 2-layer deflection d_2 and
- the deflection d_1 that describes the stiffness of the subgrade.

In this evaluation, we can use the fact that we know what exactly material is used as the base and what exactly is the subgrade. Based on this information, we can determine the mechanical characteristics of the subgrade and of the base, such as the values of the parameters k_2' and k_3' that describe the non-linearity of the corresponding materials according to the model

$$E = k_1' \cdot \left(\frac{\theta}{P_a} + 1\right)^{k_2'} \cdot \left(\frac{\tau_{\rm oct}}{P_a} + 1\right)^{k_3'}$$

that described how the elastic modulus E depends on the bulk stress

$$\theta = \sigma_1 + \sigma_2 + \sigma_3$$

and on the octahedral shear stress

$$\tau_{\rm oct} = \frac{1}{3} \cdot \sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_1 - \sigma_3)^2 + (\sigma_2 - \sigma_3)^2};$$

here, σ_1, σ_2, and σ_3 are principal stresses. This model was proposed in [57], and in [44], it was shown to be the most adequate model for describing the resilient stress.

A natural measure of the pavement stiffness is the corresponding *elastic modulus*. This modulus is used in pavement design. Because of the non-linearity of the actual pavement mechanics – as described by the above formula – the elastic modulus has different values at different depths. To gauge the quality of the pavement, it is therefore desirable to use the "average" (representative) modulus, that reflects the stiffness of the base as a whole. A reasonable idea is therefore to use the modulus at half-depth of the base as this representative stiffness.

Summarizing: by using

- the deflection d_1 of the subgrade and
- the deflection d_2 of the 2-layer pavement,

we need to estimate the representative modulus E of the base. In this estimation, we can use the values k_2' and k_3' corresponding to the subgrade and to the base.

In contrast to the *forward problem*, when we know the mechanical properties of both layer and we need to estimate the resulting deflections, the problem of reconstructing the material properties of the base from the observed deflections is known as a the *inverse problem*.

2.2 Our General Approach to Solving the Inverse Problem

To test different methods of solving the inverse problem, we performed numerous simulations of the forward problem, i.e.:

- the simulations of the subgrade and
- the simulations of the 2-layer pavements.

Both simulations were performed for different combinations of parameters describing the mechanical properties of the two layers.

Since simulating the full dynamical non-linear behavior of the pavement is very time-consuming, we performed the simulations on several simplified pavement models. Our expectation was that some of these simplified models will be accurate enough to describe the true behavior of the pavement. Specifically:

- We started with *linear static models*, in which the modulus is assumed to have the same value within each layer.
- Next, we performed simulations based on *non-linear static models*, where we take into account the above formula describing the dependence of the modulus on the depth.
- Finally, we also performed the *dynamical simulations.*

Specifically:

- First, we used the linear static simulations to find a method for solving the inverse problem in this approximation.
- Then, we modified this method so that it will be able to solve the non-linear static case.
- After that, the method was further modified to accommodate the simulation results that take the dynamic character of the problem into account.

2.3 Results of Our Analysis: Static Case

1-layer static case. For the 1-layer case, when we only have the subgrade, the representative modulus E_1 can be estimated based on the deflection d_1: $E_1 = \dfrac{c}{d_1}$, for a constant $c \approx 209$; see Section 2.5.1 for the derivation of this formula.

2-layer static case. We started by analyzing the results of linear static and nonlinear static simulations. In these two cases, we have found analytical formulas that compute the modulus E_2 of the base from the deflections d_1 and d_2.

Static linear case. In this case, the modulus E_2 can be obtained by the following formula:

$$E_2 = \frac{1}{d_2} \cdot \exp\left(a(h) + (\ln(c) - a(h)) \cdot \frac{d_1}{d_2}\right).$$

In this formula, $c = 209$, and the coefficient $a(h)$ depends on the thickness h of the base:

- for $h = 6$ inches, we have $a(h) = 1.89$;
- for $h = 12$ inches, we have $a(h) = 3.82$;
- for $h = 18$ inches, we have $a(h) = 4.66$.

The detailed derivation of this formula is given in Section 2.5.1.

Static stationary nonlinear case: general description. In this case, we need to find the representative modulus E of the base based on the following information:

- the displacement d_2 of the 2-layer pavement;
- the values k'_2 and k'_3 corresponding to the base; these values are denoted by k'_{2b} and k'_{3b}; and
- the information about the subgrade.

In the ideal case, we have as much information as possible about the subgrade; namely:

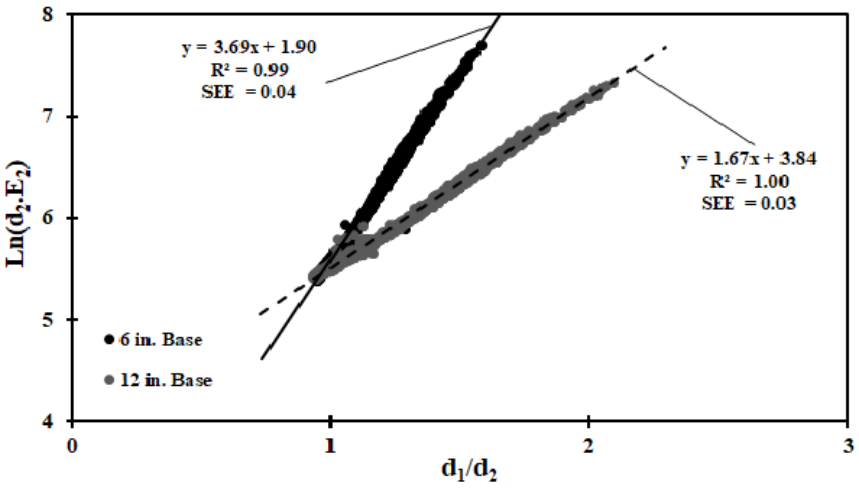

Figure 2.1: Linear static case

- the displacement d_1 of the subgrade; and
- the values k'_2 and k'_3 corresponding to the subgrade; these values are denoted by k'_{2s} and k'_{3s}.

For this case, we have come up with the following formulas that describe the representative modulus E of the base.

Static stationary nonlinear case: 150 mm base (case of full information about the subgrade). For the 150 mm cases, we have

$$\ln(d_2 \cdot E) = 2.098 + 0.361 \cdot k'_{2b} + 0.336 \cdot k'_{3b} +$$

$$0.093 \cdot k'_{2b} \cdot k'_{3b} + 0.053 \cdot (k'_{3b})^2 + 0.467 \cdot (k'_{2s}) - 0.305 \cdot (k'_{2s})^2 - 0.264 \cdot k'_{2s} \cdot k'_{3s} - 0.079 \cdot (k'_{3s})^2 +$$

$$0.242 \cdot k'_{2b} \cdot k'_{2s} + 0.091 \cdot k'_{2b} \cdot k'_{3s} + 0.053 \cdot k'_{3b} \cdot k'_{2s} +$$

$$3.509 \cdot \frac{d_1}{d_2} - 0.955 \cdot \left(\frac{d_1}{d_2} - 1 \right)^2 .$$

The R^2 is 0.95, and the mean square accuracy of this approximation is 16%.

Static stationary nonlinear case: 300 mm base (case of full information about the subgrade). For the 300 mm cases, we have

$$\ln(d_2 \cdot E) = 3.870 + 0.380 \cdot k'_{2b} + 0.348 \cdot k'_{3b} + 0.408 \cdot k'_{2s} + 0.196 \cdot k'_{3s} +$$

$$0.078 \cdot k'_{2b} \cdot k'_{3b} + 0.037 \cdot (k'_{3b})^2 - 0.177 \cdot (k'_{2s})^2 - 0.160 \cdot k'_{2s} \cdot k'_{3s} - 0.029 \cdot (k'_{3s})^2 +$$

$$0.138 \cdot k'_{2b} \cdot k'_{2s} + 0.065 \cdot k'_{2b} \cdot k'_{3s} + 0.069 \cdot k'_{3b} \cdot k'_{2s} + 0.041 \cdot k'_{3b} \cdot k'_{3s} +$$

$$1.656 \cdot \frac{d_1}{d_2} - 0.294 \cdot \left(\frac{d_1}{d_2} - 1\right)^2.$$

The R^2 is 0.96, and the mean square accuracy of this approximation is 11%.

Comment. The detailed derivation of these formulas is given in Section 2.5.2.

What if we only know the estimate of the subgrade's representative modulus. As we have mentioned in the beginning of this section, to estimate the representative modulus E_1 of the subgrade, we do not really need to know the values of the parameters k'_2 and k'_3 corresponding to the subgrade: it is sufficient to know the corresponding displacement d_1. As a result, practitioners do not need to estimate these parameters when compacting the subgrade.

It is therefore reasonable to also consider a realistic scenario in which instead of the values d_1, k'_{2s}, and k'_{3s} corresponding to the subgrade, we only know the estimate E_1 that we obtained when we compacted the subgrade.

Another reason why such a scenario is needed is that while, in principle, we could estimate the values of these parameters based on our knowledge of the subgrade material: gravel, sand, clay, etc., but from the viewpoint of accuracy, there is a big difference between the base and the subgrade: for the base, we prepare the material, while the subgrade is largely the compressed soil. The properties of the base are thus reasonably well known, while the properties of the subgrade may vary greatly from site to site. As a result, practitioners often do not place much weight on the accuracy and reliability of such estimates.

In this scenario, in addition to the displacement d_2, and the parameters h, k'_{2b}, and k'_{3b} that describe the base, we have an estimate E_1 for the representative modulus of the

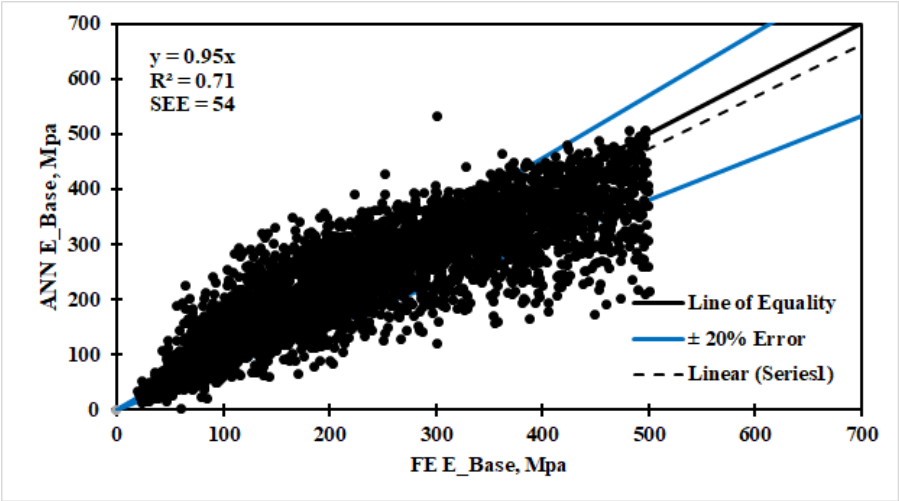

Figure 2.2: Case when we only know an estimate for E_1

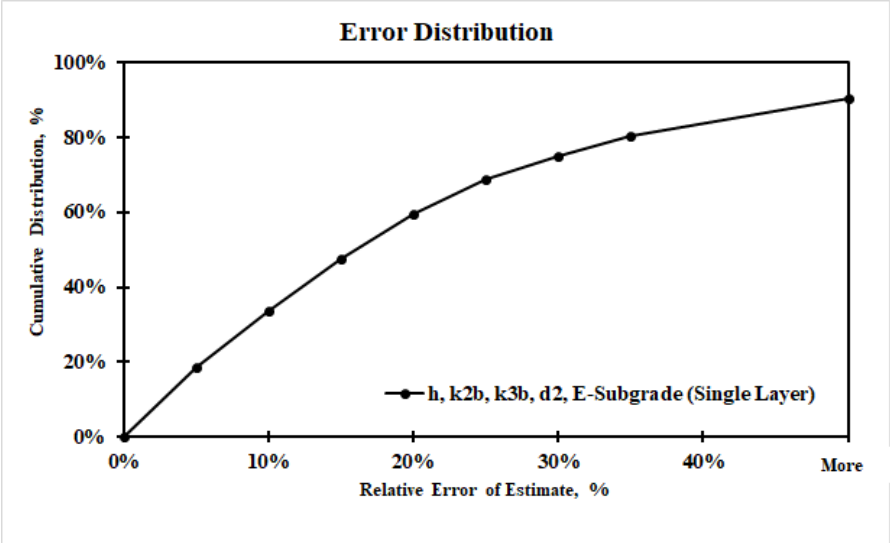

Figure 2.3: Case when we only know an estimate for E_1

subgrade. To estimate the modulus E_2 in this scenario, we trained a neural network; the results of this neural network are represented in Fig. 2.2 and Fig. 2.3.

Comment. Slightly more accurate estimates can be obtained if we take into account that the above formula for the dependence of E_1 on d_1, while reasonably accurate, is still approximate. Thus, we can get better estimates if, instead of using the estimate E_1, we use

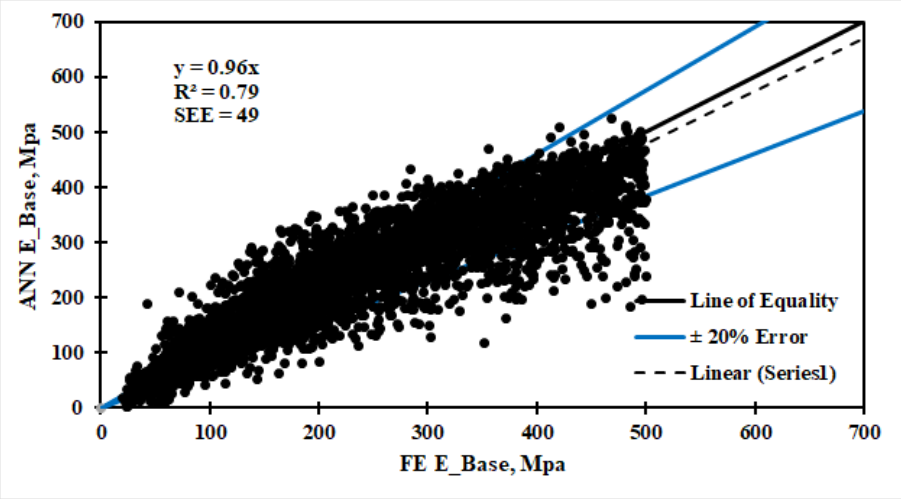

Figure 2.4: Case when we use d_1 instead of E_1

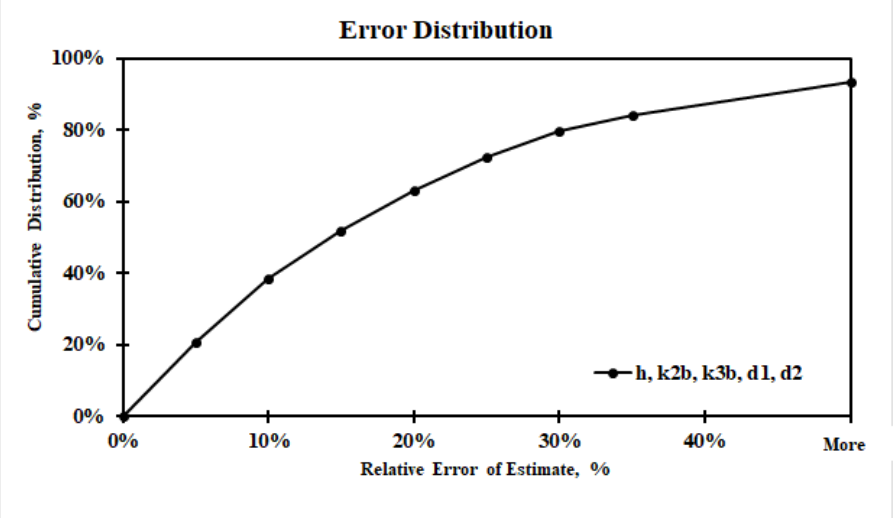

Figure 2.5: Case when we use d_1 instead of E_1

the displacement d_1 – this way, we avoid the effect of the above inaccuracy.

In other words, as inputs for estimating E_2, we use d_2, h, k'_{2b}, k'_{3b}, and the displacement d_1. The result of training the corresponding neural network model resulting estimates are presented in Fig. 2.4 and Fig. 2.5.

Even more accurate estimates for E_2 can be obtained if we use the actual values E_1^{act} of

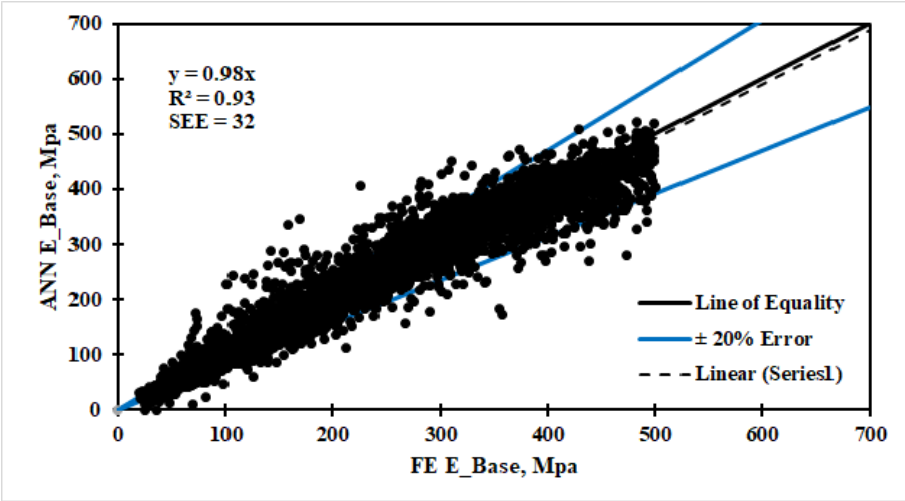

Figure 2.6: Case when we use the actual value E_1^{act}

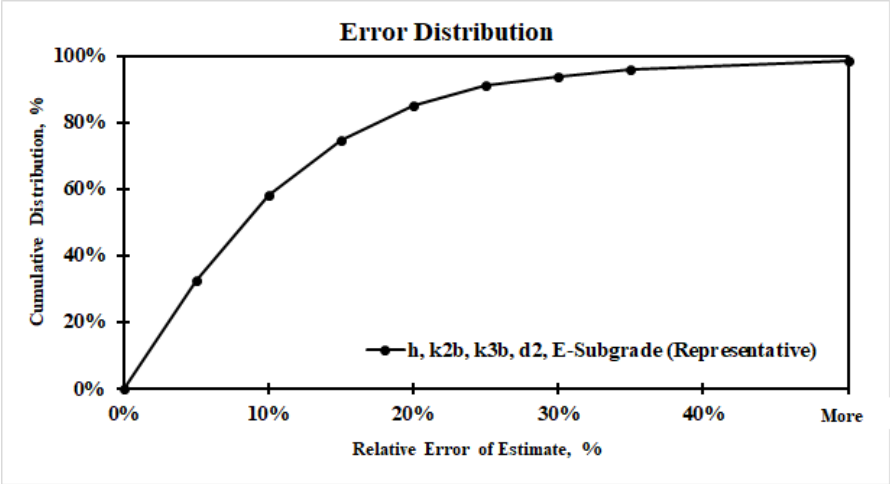

Figure 2.7: Case when we use the actual value E_1^{act}

the subgrade's representative modulus E_1 instead of the value d_1 or the d_1-based estimate for E_1. In this case, to estimate E_2, we use the values d_2, h, k'_{2b}, k'_{3b}, and E_1^{act}. The results of the corresponding neural network model are presented in Fig. 2.6 and Fig. 2.7.

To make comparison between these three neural network models easier, we combine the distribution of the estimation errors of all three models in a single graph; see Fig. 2.8.

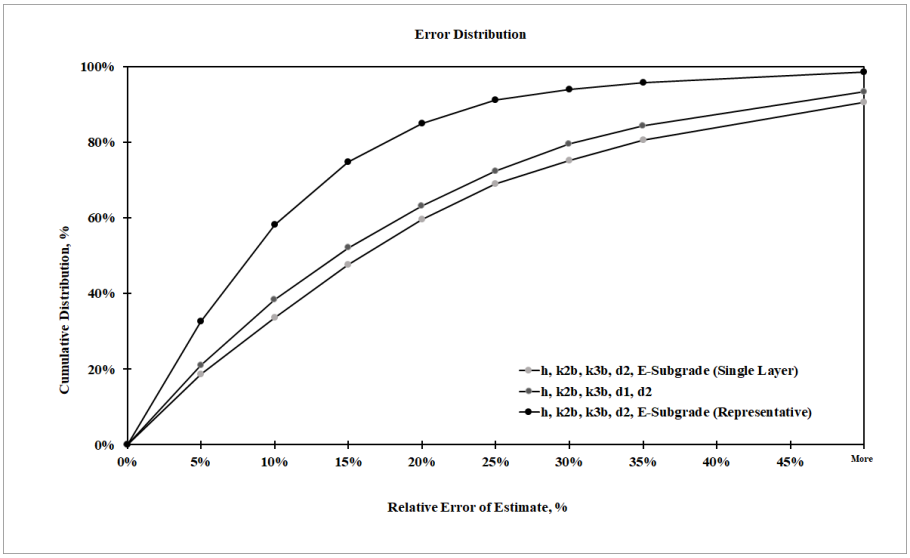

Figure 2.8: Comparative accuracy of three neural network models

2.4 Analysis of the Dynamic Case: Dynamic Case Can Be Reduced to Static Case

Desired accuracy. All the simulations are based on the simplifying assumption that both the subgrade and the base are homogeneous. In practice, the mechanical properties of the subgrade and of the base can randomly change by 20-25% within the same road segment. Because of this, even if we know the average values of the parameters characterizing the base and the subgrade, it is not possible to predict the mechanical properties of the pavement at a given location with accuracy better than 20-25%.

From this viewpoint, if we have two models for predicting the mechanical properties of the pavement:

- a simplified easier-to-compute model and
- a more realistic model that requires much more computation time,

and if the predictions of these two models differ by less than 20-25%, this means that from the practical viewpoint, the simplified easier-to-compute model is good enough – and there

is no practical need for more sophisticated computational models.

Is it necessary to perform dynamic simulations: formulation of the problem. We started with the simplified easy-to-compute static model. In reality, the corresponding processes are dynamic, so theoretically, to get a more accurate description, we need to take the dynamic character of the processes into account and perform dynamic computations. However, dynamic computations require an order of magnitude more computations than the static ones.

Since our ultimate objective is to come up with formulas that will be easy to apply in the field, we would like the resulting model to be as easy-to-compute as possible. In view of the above-mentioned inaccuracies caused by inhomogeneity, a natural question is: do we really need to perform dynamic computations? If by using static computations, we can get the same results as with dynamic ones within the desired 20-25% accuracy, this would mean that static computations are sufficient.

To check on this, we compared the results of static and dynamic simulations.

Comparing the results of static and dynamic simulations: 1-layer case. We started by comparing the results of static and dynamic simulations on the 1-layer case, when we only have the subgrade.

The elastic modulus of the subgrade is usually estimated by plugging in the standard values of θ and τ_{oct} into the general formula. It is assumed that, based on the structure of the subgrade, we know the corresponding values of k'_2 and k'_3, so the only parameter that we need to determine to find out the elastic modulus is k'_1.

We need to determine this parameter k'_1 based on the deflection d_1. In the static case, to find the elastic modulus E, we had a formula for the expression $\ln(d_1 \cdot E)$ (see derivation in Section 2.5) – and once we know this expression and we know d_1, we can therefore determine E.

Let us therefore look for formulas that describe a similar expression $\ln(d_1 \cdot k'_1)$. For the same values k'_1, k'_2, and k'_3, we have two different deflections d_1:

- the deflection d_1^{stat} based on the static computations, and
- the deflection d_1^{dyn} based on the dynamic computations.

Thus, we have two different expressions:

- the expression $\ln(d_1^{\text{stat}} \cdot k_1')$ based on the static computations, and
- the expression $\ln(d_1^{\text{dyn}} \cdot k_1')$ based on the dynamic computations.

The question is: once we know $\ln(d_1^{\text{stat}} \cdot k_1')$, can we reconstruct $\ln(d_1^{\text{dyn}} \cdot k_1')$ with the desired accuracy?

To check on this, we first tried to use linear regression, and to predict $\ln(d_1^{\text{dyn}} \cdot k_1')$ as a linear function of $\ln(d_1^{\text{stat}} \cdot k_1')$:

$$\ln(d_1^{\text{dyn}} \cdot k_1') = a_0 + a_1 \cdot \ln(d_1^{\text{stat}} \cdot k_1').$$

It turns out that we indeed have this dependence, with $a_0 = 0.67$ and $a_1 = 0.91$. The resulting mean square error is 14% – which is much smaller than the desired 20-25% threshold.

Interestingly, when we try to add all possible terms linear or quadratic in terms of k_2' and k_3' this dependence (i.e., terms proportional to k_2', k_3', $(k_2')^2$, $(k_3')^2$, and $k_2' \cdot k_3'$), the coefficients at all these new terms were statistically insignificant ($p > 0.01$).

Thus, in the 1-layer case, the results of dynamic simulations can indeed be easily reconstructed from the results of static simulations – and thus, in the 1-layer case, the time-consuming dynamic simulations are not needed.

2-layer case: which values to choose. For the dynamic case, the elastic modulus depends not only on the location, but also on time. In our simulations, we simulated the first six cycles of the corresponding vibrations. In these six cycles, we ended up with slightly different values of the elastic modulus.

Similarly to how we decided to select the spatially-average value of the elastic modulus as the most representative one, it is reasonable to select the time-average value of the elastic

modulus. Thus, as the representative modulus of the base, we selected the arithmetic average of the values corresponding to all six cycles.

Comment. It is important to take into account that the main objective of the compaction process is to compact the pavement, i.e., to improve the mechanical properties of the pavement. Thus, in the real-life compaction process, the parameters k_1', k_2', and k_3' that describe the mechanical properties of the base and of the subgrade, change with time until they stabilize. The short last stage of the compaction process – when the pavement has already been compacted and its mechanical properties have already stabilized – is therefore the only stage at which the parameters k_1', k_2', and k_3' do not change with time.

In our dynamic simulations, we assume that the parameters k_1', k_2', and k_3' do not change with time. Thus, our simulations corresponds to the above-mentioned last stage of the compaction process, when the pavement has already been compacted. The six cycles of our simulation – over which we average – thus correspond to the *last* 6 cycles of the actual compaction process.

From this viewpoint, the average of the elastic modulus values corresponding to all 6 cycles of the simulation corresponds to selecting the average of the last six cycles of the compaction process.

How to compare static and dynamic cases. In many practical applications, the approximation accuracy is gauged by the mean square approximation error

$$\sigma \stackrel{\text{def}}{=} \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} (\Delta y_i)^2},$$

where n is the number of approximated values and Δy_i is the approximation error of the i-th experiment. This estimation method works well when all the approximating values are close to the actual values (that they are approximating), and thus, all approximations errors are small.

However, in the same dynamic simulation, the values of the elastic modulus corresponding to 6 cycles differ by a factor of 10. From the statistical viewpoint, when we have such

large differences, the traditional mean square error does not provide us with an adequate measure of accuracy – since the mean square error will be influenced by such "outliers"; see, e.g., [33]. In such situations, statistics recommends to use *robust* methods for estimating the accuracy. The most widely use robust measure for estimating accuracy is the mean absolute value of the absolute error, i.e., the Mean Absolute Error (MAE) $\frac{1}{n} \cdot \sum_{i=1}^{n} |\Delta y_i|$ [33]. This is the value that we will use to gauge how close are the results of static and dynamic simulations.

Comparing the results of static and dynamic simulations: 2-layer case. In our analysis, similar to the static case, we consider two typical thicknesses of the base: 150 mm and 300 mm. In both cases, we want to check whether the expression $\ln(d_2^{\rm dyn} \cdot E^{\rm dyn})$ correspond to the dynamic case can be described in terms of the similar static expression $\ln(d_2^{\rm stat} \cdot E^{\rm stat})$ (with a possible influence of the parameters k_2' and k_3' characterizing the base and the subgrade).

For the 150 mm case, the formula

$$\ln(d_2^{\rm dyn} \cdot E^{\rm dyn}) = a_0 + a_1 \cdot \ln(d_2^{\rm stat} \cdot E^{\rm stat}),$$

with $a_0 = 0.96$ and $a_1 = 0.81$, predicts the results of the dynamic simulations with the 17% accuracy. For the 300 mm case, a similar formula with $a_0 = 1.68$ and $a_1 = 0.69$ provides a even better 16% accuracy.

In both cases, the mean approximation error is below the 20% threshold. So, in the 2-layer case as well, the results of dynamic simulations can be easily reconstructed from the results of static simulations – and thus, in the 2-layer case, the time-consuming dynamic simulations are not needed.

Comment. It is worth mentioning that we get an even better approximation accuracy if we take into account possible non-linear terms in the dependence of the dynamic expression on the static one, i.e., if we consider an approximation formula

$$\ln(d_2^{\rm dyn} \cdot E^{\rm dyn}) = a_0 + a_1 \cdot \ln(d_2^{\rm stat} \cdot E^{\rm stat}) +$$

$$a_2 \cdot (\ln(d_2^{\text{stat}} \cdot E^{\text{stat}}))^2 + a_3 \cdot (\ln(d_2^{\text{stat}} \cdot E^{\text{stat}}))^3.$$

2.5 Derivation of the Formulas

2.5.1 Solving Inverse Problem for the Static Linear Case

Linear elastic equations: reminder. Linear elastic equations have the form

$$\sigma_{ij,j} + F_i = 0, \tag{A1}$$

$$\varepsilon_{ij} = \frac{1}{E} \cdot ((1+\nu) \cdot \sigma_{ij} - \nu \cdot \delta_{ij} \cdot \sigma_{kk}), \tag{2.2}$$

and

$$\varepsilon_{ij} = \frac{1}{2} \cdot (u_{i,j} + u_{j,i}), \tag{2.3}$$

where, for each quantity a, $a_{,i}$ means partial derivative of a with respect to x_i:

$$a_{,i} \stackrel{\text{def}}{=} \frac{\partial a}{\partial x_i}.$$

The equation relating ε_{ij} and σ_{ij} can be equivalently rewritten as

$$E \cdot \varepsilon_{ij} = (1+\nu) \cdot \sigma_{ij} - \nu \cdot \delta_{ij} \cdot \sigma_{kk}. \tag{2.22.}$$

In our forward simulations:

- the force F is fixed, and
- the Poisson ratio ν is fixed.

We:

- select the moduli E for the two layers, and
- compute the deflections u_i.

1-layer case: analysis of the problem. In the 1-layer case, we assume that the modulus E has the same value for all spatial locations and for all depths. Suppose that u_i is a solution corresponding to this modulus E. One can easily show that if we select a different modulus E', then we get deflections

$$u_i' = \frac{E}{E'} \cdot u_i \tag{2.4}$$

Indeed, in this case, due to formula (3), we have $\varepsilon_{ij}' = \dfrac{E}{E'} \cdot \varepsilon_{ij}$, hence $E' \cdot \varepsilon_{ij}' = E \cdot \varepsilon_{ij}$. Thus, the equation (2.2a) – and hence, all three equations (2.1), (2.2a), and (2.3) – is satisfied when we take the new deflections u_i' and the same stress σ_{ij} as before: $\sigma_{ij}' = \sigma_{ij}$.

1-layer case: conclusion. In the 1-layer case, the product of the displacement and the modulus should be the same for all the values of the subgrade modulus. So, if for different values of the modulus E_1, we multiply this value by the displacement d_1, we should get a constant $d_1 \cdot E_1 = \text{const}$, and

$$E_1 = \frac{c}{d_1}$$

for this constant c.

Indeed, for the results of numerical simulations, the product $d_1 \cdot E_1$ is almost the same, it ranges from 204.5 to 211.5.

The 3-4% difference between the theoretical prediction and the results of numerical simulations is clearly caused by the fact that we are using approximate finite-element techniques to solve the elasticity equations. As c, we can thus take an average value of this product $c \approx 209$.

2-layer case: general idea. In the 2-layer case, the ratio $d_2 \cdot E_2$ is no longer a constant, it may depend on d_1 and d_2:

$$d_2 \cdot E_2 = f(d_1, d_2). \tag{2.5}$$

In the 2-layer case, arguments similar to the ones given in the analysis of the 1-layer case show that if we multiply both moduli E_1 and E_2 by the same constant k, i.e., take $E_1' = k \cdot E_1$ and $E_2' = k \cdot E_2$, the displacement d_2 should then divide by the same constant:

$d_2' = \dfrac{1}{k} \cdot d_2$. In this case, the product $d_2 \cdot E_2$ remains the same: $d_2' \cdot E_2' = d_2 \cdot E_2$. Thus, the formula (2.5) and a similar formula relating d_2' and E_2' imply that for all d_1, d_2, and k, we have

$$f(d_1, d_2) = f(d_1', d_2') = f\left(\frac{d_1}{k}, \frac{d_2}{k}\right). \tag{2.6}$$

In particular, for $k = d_2$, we conclude that

$$f(d_1, d_2) = f\left(\frac{d_1}{d_2}, 1\right). \tag{2.7}$$

When $E_1 = E_2$, then the two layers have the exact same mechanical properties, i.e., in effect, we have the 1-layer case. In this case, we have $d_1 = d_2$, and we have $d_2 \cdot E_2 = c$, hence $f(d_1, d_1) = c$. So, we can rewrite the formula (2.7) as

$$d_2 \cdot E_2 = f(d_1, d_2) = g(r), \tag{2.8}$$

where we denoted

$$r \stackrel{\text{def}}{=} \frac{d_1}{d_2}$$

and $g(r) \stackrel{\text{def}}{=} f(r, 1)$. The advantage of this equivalent form is that for $d_1 = d_2$, i.e., for $r = 1$, we have $d_2 \cdot E_2 = c$ and thus, $g(1) = c$.

2-layer case: empirical analysis. For each simulation of the 12 inch case, we computed the value of the product $d_2 \cdot E_2$ and the ratio. To get a general idea of how the product depends on r, we looked at the cases when $d_2 \cdot E_2$ is approximately equal to 400, to 800, and to 1600. For these values, we got $r \approx 1.3$, $r \approx 1.7$, and $r \approx 2.1$. We noticed that the increase from each case to the next one is the same: $1.7 - 1.3 = 2.1 - 1.7 = 0.4$. So, when we double the product, the value r increases by the same amount.

There is a known function that has this property: logarithm. Indeed, since the logarithm of the product is equal to the sum of the logarithm, we have $\log(2x) = \log(x) + \log(2)$. So, when we double the input to the logarithm, the result gets increased by the same constant $\log(2)$.

A similar property holds for any linear function of the logarithm, i.e., for any function of the type $F(x) \stackrel{\text{def}}{=} c_1 + c_2 \cdot \log(x)$. Indeed,

$$F(2x) = c_1 + c_2 \cdot \log(2x) = c_1 + c_2 \cdot (\log(x) + \log(2)) =$$

$$c_1 + c_2 \cdot \log(x) + c_2 \cdot \log(2) = (c_1 + c_2 \cdot \log(x)) + c_2 \cdot \log(2) = F(x) + c_2 \cdot \log(2),$$

i.e., $F(2x) = F(x) + c. \log(2)$. So, if we double x, the value of $F(x)$ is increased by an additive constant $c_2 \cdot \log(2)$.

Resulting conjecture. We therefore conjectured that r is a linear function of the logarithm $\ln(d_2 \cdot E_2)$, or, equivalently, that the logarithm $\ln(d_2 \cdot E_2)$ should linearly depend on r:

$$\ln(d_2 \cdot E_2) = a + b \cdot r = a + b \cdot \frac{d_1}{d_2} \tag{2.9}$$

for some values a and b.

For $d_1 = d_2$, the formula (2.9) implies that $\ln(c) = a + b$, so we can conclude that $b = \ln(c) - a$ and thus,

$$\ln(d_2 \cdot E_2) = a + (\ln(c) - a) \cdot \frac{d_1}{d_2}. \tag{2.10}$$

This formula has been confirmed. It turns out that this formula perfectly describes all the simulation results.

We checked this formula on the simulations corresponding to the two most frequently used base thicknesses: $h = 12$ inches and $h = 6$ inches. We also tested in on the case of $h = 18$ inches. For all three thicknesses, we get a perfect first with the results of >200 simulations corresponding to each thickness value.

Conclusion. If we know the deflections d_1 and d_2, and the thickness h, then, from the formula (2.11), we can determine the modulus E_2 of the base as

$$E_2 = \frac{1}{d_2} \cdot \exp\left(a(h) + (\ln(c) - a(h)) \cdot \frac{d_1}{d_2}\right). \tag{2.12}$$

2.5.2 Solving Inverse Problem for the Static Non-Linear Case

Let us start with the 1-layer case. Similarly to the linear static case, let us start with the 1-layer case, when we only have a subgrade.

In this case, we know the deflection d_1, we know the parameters k'_2 and k'_3 characterizing the subgrade, and we want to find the stiffness of the subgrade.

As a measure of this stiffness, it is reasonable to take a representative value

$$E = k'_1 \cdot \left(\frac{\theta}{P_0} + 1\right)^{k'_2} \cdot \left(\frac{\tau_{\text{oct}}}{P_0} + 1\right)^{k'_3} \tag{2.131}$$

for standard θ and τ_{oct}. Since we know the values k'_2, k'_3, θ, and τ_{oct}, the only parameter that we need to determine the representative modulus is k'_1. Let us analyze how we can determine this parameter.

As we have mentioned in Subsection 2.5.1, in the static linear case, the elastic modulus E is related to the corresponding displacement d_1 by the formula

$$\ln(d_1 \cdot E) = \text{const}. \tag{2.14}$$

Substituting the expression (2.13) for the elastic modulus into the formula (2.14), we conclude that

$$\ln\left(d_1 \cdot k'_1 \cdot \left(\frac{\theta}{P_0} + 1\right)^{k'_2} \cdot \left(\frac{\tau_{\text{oct}}}{P_0} + 1\right)^{k'_3}\right) = \text{const}.$$

Taking into account that the logarithm of the product is equal to the sum of the logarithms and that the logarithm of the power is equal to $\ln(a^b) = b \cdot \ln(a)$, we conclude that

$$\ln(d_1 \cdot k'_1) + a_2 \cdot k'_2 + a_3 \cdot k'_3 = \text{const}, \tag{2.15}$$

where we denoted $a_2 \stackrel{\text{def}}{=} \ln\left(\frac{\theta}{P_0} + 1\right)$ and $a_3 \stackrel{\text{def}}{=} \ln\left(\frac{\tau_{\text{oct}}}{P_0} + 1\right)$. Thus, in the static linear case, we have

$$\ln(d_1 \cdot k'_1) = \text{const} - a_2 \cdot k'_2 - a_3 \cdot k'_3. \tag{2.16}$$

In other words, in the linear static case, the expression $\ln(d_1 \cdot k'_1)$ is a linear function of the parameters k'_2 and k'_3.

In many cases – e.g., when the subgrade is sufficiently stiff – the linear model is a good approximation. It is therefore reasonable to look for models in which we add smaller size terms to the main (linear) terms. A natural class of such models are polynomial models, where to linear terms, we add quadratic terms, then, if needed, cubic terms, etc.

Let us start with the quadratic terms. In general, adding quadratic terms means that we are looking for the dependence of the type

$$\ln(d_1 \cdot k'_1) = c_0 + c_2 \cdot k'_2 + c_3 \cdot k'_3 + c_{22} \cdot (k'_2)^2 + k'_{23} \cdot k'_2 \cdot k'_3 + c_{33} \cdot (k'_3)^2. \tag{2.17}$$

2-layer case. In the nonlinear case, to predict the value $\ln(d_2 \cdot E)$, in addition to the ratio $\dfrac{d_1}{d_2}$ that we used in the linear case, it is reasonable to also use terms quadratic in terms of the parameters k'_2 and k'_3 – as in the formula (2.17) describing the 1-layer case.

In the 1-layer case, we had two parameters k'_2 and k'_3. In the 2-layer case, when we have the base and the subgrade, we thus have four parameters:

- the two parameters k'_{2b} and k'_{3b} of the base, and
- the two parameters k'_{2s} and k'_{3s} of the subgrade.

So, if we add, to the linear dependence on $\dfrac{d_1}{d_2}$, all possible terms quadratic in terms of the parameters k'_2 and k'_3 of the base and of the subgrade, we get the following formula

$$\ln(d_2 \cdot E) = c_0 + c_{2b} \cdot k'_{2b} + c_{3b} \cdot k'_{3b} + c_{2s} \cdot k'_{2s} + c_{3s} \cdot k'_{3s} +$$

$$c_{22b} \cdot (k'_{2b})^2 + c_{23b} \cdot k'_{2b} \cdot k'_{3b} + c_{33b} \cdot (k'_{3b})^2 + c_{22s} \cdot (k'_{2s})^2 + c_{23s} \cdot k'_{2s} \cdot k'_{3s} + c_{33s} \cdot (k'_{3s})^2 +$$

$$c_{2b2s} \cdot k'_{2b} \cdot k'_{2s} + c_{2b3s} \cdot k'_{2b} \cdot k'_{3s} + c_{3b2s} \cdot k'_{3b} \cdot k'_{2s} + c_{3b3s} \cdot k'_{3b} \cdot k'_{3s} +$$

$$c_{1d} \cdot \frac{d_1}{d_2}. \tag{2.18}$$

In principle, we can also have quadratic terms that include $\dfrac{d_1}{d_2}$; such terms are:

$$\left(\frac{d_1}{d_2} - 1\right)^2, \quad k'_{2b} \cdot \left(\frac{d_1}{d_2} - 1\right), \quad k'_{3b} \cdot \left(\frac{d_1}{d_2} - 1\right),$$

$$k'_{2s} \cdot \left(\frac{d_1}{d_2} - 1\right), \quad k'_{3s} \cdot \left(\frac{d_1}{d_2} - 1\right). \tag{2.19}$$

We used the linear regression program available with the Excel Data Analysis tool to find the coefficients c_i that provide the best fit for the observed values $d_2 \cdot E$. This tool not only provides us with the best fit values of the coefficients, it also supplies, for each of the coefficients, the p-value describing to what extent this particular variable is relevant for predicting the desired value (in our case, the value of $\ln(d_2 \cdot E)$). It is then reasonable to only keep the coefficients for which the corresponding p-value is smaller than the given threshold. In this analysis, we considered the usual threshold of 0.01.

The Excel tool allows us to use only up to 16 parameters, so we started with all the terms from the formula (2.18). Then, we deleted terms for which the p-value was above our threshold 0.01, and replaced them with one of the remaining quadratic terms, after which we repeated this process.

Both for the 150 mm and for the 300 mm cases, the only significant quadratic term containing $\frac{d_1}{d_2}$ turned out to be the term $\left(\frac{d_1}{d_2} - 1\right)^2$, all other terms from the list (2.19) turned out to be insignificant.

In both cases, we also eliminated the insignificant term proportional to $(k'_{2b})^2$. This makes sense: non-linearity is most important for soft materials, and the base is much stiffer than the subgrade. So, it makes sense that nonlinear terms corresponding to the base are much less significant that the terms corresponding to the subgrade.

In the 150 mm cases, also terms proportional to k'_{3s} and $k'_{3b} \cdot k'_{3s}$ turned to be insignificant. As a result, we got the formulas presented in the main text.

2.5.3 Static vs. Dynamic: What If We Use Mean Square Error

For 150 mm, we started by trying to describe $\ln(d_2^{\rm dyn} \cdot E^{\rm dyn})$ as a linear function of $\ln(d_2^{\rm stat} \cdot E^{\rm stat})$. We got a good match, but with the mean square error of 29%, which is higher than the desired threshold of 20-25%.

If we allow terms which are quadratic and cubic in terms of $\ln(d_2^{\rm stat} \cdot E^{\rm stat})$, we decrease

the mean square error to 28% – still above the threshold. A peculiar thing about this was that the p-values for all the terms were greater than 0.01. If we follow the standard statistical procedures in this case, we would need to delete all the terms – which makes no sense. So, in this case, we did not dismiss terms with large p-values.

When we added, to the terms proportional to

$$\ln(d_2^{\text{stat}} \cdot E^{\text{stat}}), \quad (\ln(d_2^{\text{stat}} \cdot E^{\text{stat}}))^2, \text{ and } (\ln(d_2^{\text{stat}} \cdot E^{\text{stat}}))^3,$$

terms which are linear in k'_{2b}, k'_{3b}, k'_{2s}, k'_{3s}, and $d_1/d_2 - 1$, then we got to the desired 25% accuracy.

For the 300 mm case, a similar analysis lead to 18% accuracy. In this case, the only terms that we need to add to terms depending only on $\ln(d_2^{\text{stat}} \cdot E^{\text{stat}})$ where the terms proportional to $(k'_{2b})^2$ and $k'_{2b} \cdot k'_{3b}$.

Chapter 3

How to Estimate Elastic Modulus for Unbound Aggregate Materials: A Theoretical Explanation of an Empirical Formula

To ensure the quality of pavement, it is important to make sure that the elastic moduli – that describe the stiffness of all the pavement layers – exceed a certain threshold. From the mechanical viewpoint, pavement is a non-linear medium. Several empirical formulas have been proposed to describe this non-linearity. In this chapter, we describe a theoretical explanation for the most accurate of these empirical formulas.

Comment. The contents of this chapter was published in [6].

3.1 Formulation of the Problem

Need for estimating elastic modulus. To ensure the quality of a road, it is important to make sure that all the pavement layers have reached a certain stiffness level. To characterize stiffness of unbound pavement materials, transportation engineers use *elastic modulus* E.

The corresponding elastic modulus differs from the usual modulus of elasticity:

- the usual modulus corresponds to a *slowly* applied load, while
- the desired elastic modulus characterizes the effect of *rapidly* applied loads – like

those experienced by pavements.

Need to take non-linearity into account. In the usual (*linear*) elastic materials, the modulus does not depend on the stress value. In contrast, pavement materials are usually *non-linear*, in the sense that the elastic modulus non-linearly depends on the stress.

Empirical formulas describing pavement's non-linearity. Several empirical formulas have been proposed to describe this dependence. Experimental comparison [44] shows that the best description is provided by the formula (first proposed in [57])

$$E = k_1' \cdot \left(\frac{\theta}{P_a} + 1\right)^{k_2'} \cdot \left(\frac{\tau_{\rm oct}}{P_a} + 1\right)^{k_3'},$$

where P_a is atmospheric pressure, θ is the *bulk stress*, i.e., the trace

$$\theta = \sum_{i=1}^{3} \sigma_{ii}$$

of the stress tensor σ_{ij} (see, e.g., [66]), and

$$\tau_{\rm oct} \stackrel{\rm def}{=} \sqrt{\frac{1}{3} \cdot \sum_{ij} \sigma_{ij}^2 - \frac{1}{3} \cdot \theta^2}$$

is the *octahedral shear stress*.

In terms of the eigenvalues σ_1, σ_2, and σ_3 of the stress tensor,

$$\theta = \sigma_1 + \sigma_2 + \sigma_3$$

and

$$\tau_{\rm oct} = \frac{1}{3} \cdot \sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}.$$

What we do in this chapter. In this chapter, we provide a theoretical explanation for the above empirical formula.

This explanation uses the general idea that the fundamental physical formulas should not change if we simply changing the measuring unit and/or the starting point for the measurement scale.

Chapter outline. First, in Section 2, we briefly explain the general idea, that fundamental physical formulas should not depend on the choice of the starting point or on the choice of the measuring unit.

In Section 3, we use this general idea to describe possible dependence of the elastic modulus E on, correspondingly, the bulk stress θ and on the octahedral sheer stress $\tau_{\rm oct}$.

Finally, in Section 4, we apply similar ideas to combine the two formulas for $E(\theta)$ and $E(\tau_{\rm oct})$ into a single formula $E(\theta, \tau_{\rm oct})$ that describes the dependence of the elastic modulus on both stresses.

Comment. In our derivation, we are not using physical equation, we are only using expert knowledge – which, in this case, is formulated in terms of invariance. From this viewpoint, this chapter can be viewed as a particular case of soft computing, techniques for formalizing and utilizing expert knowledge.

3.2 General Idea: Fundamental Physical Formulas Should Not Depend on the Choice of the Starting Point or of the Measuring Unit

Main idea. Computers process numerical values of different quantities. A numerical value of a quantity depends on the choice of a measuring unit and – in many cases – also on the choice of the starting point.

For example, depending on the choice of a measuring unit, we can describe the height of the same person as 1.7 m or 170 cm. Similarly, we can describe the same moment of time as 2 pm (14.00) if we use El Paso time or 3 pm (15.00) if we use Austin time – the difference is caused by the fact that the starting points for these two times – namely midnight (00.00) in El Paso and midnight (00.00) in Austin – differ by one hour.

The choice of a measuring unit is rather arbitrary. For example, we can measure length in meters or in centimeters or in feet. Similarly, the choice of the starting point is arbitrary:

when we analyze a cosmic event, it does not matter the time of what location we use to describe it. It is therefore reasonable to require that the fundamental physical formulas not depend on the choice of a measuring unit and – if appropriate – on the choice of the starting point. We do not expect that, e.g., Newton's laws look differently if we use meters or feet.

Of course, if we change the units in which we measure one of the quantities, then we may need to adjust units of related quantities. For example, if we replace meters with centimeters, then for the formula $v = d/t$ (that describes velocity v as a ratio of distance d and time t) to remain valid we need to replace meters per second with centimeters per second when measuring velocity. However, once the appropriate adjustments are made, we expect the formulas to remain the same.

Not all physical quantities allow both changes. It should be mentioned that while most physical quantities do not have any preferred measuring unit – and thus, selection of a different measuring unit makes perfect physical sense – some quantities have a fixed starting point. For example, while we can choose an arbitrary starting point for time, for distance, 0 distance seems to be a reasonable starting point.

As a result, while the change of a measuring unit makes sense for most physical quantities, the change of a starting point only makes sense for some of them – and a physics-based analysis is needed to decide whether this change makes physical sense.

How to describe the change of a measuring unit in precise terms. If we replace the original measuring unit with a new unit which is a times smaller, then all numerical values of the measured quantity get multiplied by a: $x' = a \cdot x$.

For example, if we replace meters with centimeters – which are $a = 100$ times smaller – then the original height of $x = 1.7$ m becomes $x' = a \cdot x = 100 \cdot 1.7 = 170$ cm.

How to describe the change of the starting point in precise terms. If we replace the original starting point by a new one which is b earlier (or smaller), then to all numerical values of the measured quantity the value b is added: $x' = x + b$.

For example, if we replace El Paso time with Austin time – which is $b = 1$ hour earlier, then the original time of $x = 14.00$ hr becomes $x' = x + b = 14.00 + 1.00 = 15.00$ hr.

In general, we can change both the measuring unit and the starting point. If we first change the measuring unit and the starting point, then:

- first, the original value x first gets multiplied by a, resulting in $x' = a \cdot x$, and
- then the value b is added to the new value x', resulting in $x'' = x' + b = a \cdot x + b$.

Thus, in general, when we change both the measuring unit and the starting point, we get a linear transformation $x \to a \cdot x + b$.

3.3 How Elastic Modulus Depends on the Bulk Stress (and on the Octahedral Shear Stress)

What we do in this section. Let us first use the above idea to describe how the elastic modulus E depends on the bulk stress θ.

Which invariances makes sense in this case. As we have mentioned in the previous section,

- while the change of a measuring unit makes sense for (practically) *all* physical quantities,
- the change of the starting point only makes physical sense for *some* quantities.

Let us therefore analyze whether the change of the starting point makes sense for the elastic modulus E and for the bulk stress θ.

For the elastic modulus, there is a clear starting point $E = 0$, in which strain does not cause any stress. So, for the elastic modulus, only a change in a measuring unit makes physical sense.

In contrast, for the bulk stress, we can clearly have several choices of the starting point, choices motivated by the fact that in addition to the external stress, there is also an always-present atmospheric pressure. One possibility is to only count the external stress and thus, consider the situation in which we only have atmospheric pressure as corresponding to zero stress. Another possibility is to explicitly take atmospheric pressure into account and take the ideal vacuum no-atmospheric-pressure situation as zero stress. In the first case, we can select atmospheric pressures corresponding to different heights as different starting points.

What does it mean for the resulting formula to be independent: first approximation. For the dependence $E(\theta)$, the requirement that this dependence does not change if we change numerical values of θ means the following. For every $a > 0$ and b, the dependence in the new units $E(a \cdot \theta + b)$ has exactly the same form as in the old units – if we also appropriately re-scale E. So, we should have

$$E(a \cdot \theta + b) = c(a, b) \cdot E(\theta) \tag{3.3.1}$$

for some value c which, in general, depends on a and b.

What are the functions that satisfy this condition: analysis of the problem. Let us find all the functions $E(\theta)$ for which, for some function $c(a, b)$, the equality (3.1) holds for all x, $a > 0$, and b.

From the physical viewpoint, small changes in θ should lead to small changes in E, i.e., in mathematical terms, the dependence $E(\theta)$ should be continuous. It is known that every continuous function can be approximated, with any given accuracy, by a differentiable function (e.g., by a polynomial). Thus, without losing generality, we can safely assume that the dependence $E(\theta)$ is differentiable.

Thus, the function

$$c(a, b) = \frac{E(a \cdot \theta + b)}{E(\theta)}$$

is also differentiable, as a ratio of two differentiable functions. For $a = 1$, the formula (3.1)

takes the form

$$E(\theta + b) = c(1, b) \cdot E(\theta). \tag{3.2}$$

Differentiating both sides of formula (3.2) with respect to b and setting $b = 0$, we get

$$E'(\theta) = c \cdot E(\theta), \tag{3.3}$$

where $f'(x)$ denote the derivative, and c is the derivative of $c(1, b)$ with respect to b for $b = 0$.

The equation (3.3) can be rewritten as

$$\frac{dE}{d\theta} = c \cdot E,$$

i.e., equivalently, as

$$\frac{dE}{E} = c \cdot d\theta.$$

Integrating both sides, we get $\ln(E) = c \cdot \theta + C_0$ for some constant C_0. Thus,

$$E = A \cdot \exp(c \cdot \theta), \tag{3.4}$$

where $A \stackrel{\text{def}}{=} \exp(C_0)$.

For $b = 0$ and $a \neq 0$, the equation (3.1) takes the form

$$E(a \cdot \theta) = c(a, 0) \cdot E(\theta).$$

Substituting the expression (3.4) into this formula, we conclude that

$$A \cdot \exp(c \cdot a \cdot \theta) = c(a, 0) \cdot \exp(c \cdot \theta). \tag{3.5}$$

When $c \neq 0$, the two sides grow with θ at a different speed, so we should have $c = 0$ and $E(\theta) = \text{const}$.

Thus, the only case when the formula $E(\theta)$ is fully invariant is when we have a linear material, with $E(\theta) = \text{const}$.

Since we cannot require all the invariances, let us require only some of them. Since we cannot require invariance with respect to *all* possible re-scalings, we should require invariance with respect to *some* family of re-scalings.

If a formula does not change when we apply each transformation, it will also not change if we apply them one after another, i.e., if we consider a composition of transformations. Each shift can be represented as a superposition of many small (infinitesimal) shifts, i.e., shifts of the type $\theta \to \theta + B \cdot dt$ for some B. Similarly, each re-scaling can be represented as a superposition of many small (infinitesimal) re-scalings, i.e., re-scalings of the type $\theta \to (1+A\cdot dt)\cdot\theta$. Thus, it is sufficient to consider invariance with respect to an infinitesimal transformation, i.e., a linear transformation of the type

$$\theta \to \theta' = (1 + A \cdot dt) \cdot \theta + B \cdot dt.$$

Invariance means that the value $E(\theta')$ has the same form as $E(\theta)$, i.e., that $E(\theta')$ is obtained from $E(\theta)$ by an appropriate (infinitesimal) re-scaling $E \to (1 + C \cdot dt) \cdot E$. In other words, we require that

$$E((1 + A \cdot dt) \cdot \theta + B \cdot dt) = (1 + C \cdot dt) \cdot E(\theta), \tag{3.6}$$

i.e., that

$$E(\theta + (A \cdot \theta + B) \cdot dt) = E(\theta) + C \cdot E(\theta) \cdot dt.$$

Here, by definition of the derivative, $E(\theta + q \cdot dt) = E(\theta) + E'(\theta) \cdot q \cdot dt$. Thus, from (3.6), we conclude that

$$E(\theta) + (A \cdot \theta + B) \cdot E'(\theta) \cdot dt = E(\theta) + C \cdot E(\theta) \cdot dt.$$

Subtracting $E(\theta)$ from both sides and dividing the resulting equality by dt, we conclude that

$$(A \cdot \theta + B) \cdot E'(\theta) = C \cdot E(\theta).$$

Since $E'(\theta) = \dfrac{dE}{d\theta}$, we can separate the variables by moving all the terms related to E to one side and all the terms related to θ to another side. As a result, we get

$$\frac{dE}{E} = C \cdot \frac{d\theta}{A \cdot \theta + b}.$$

Degenerate cases when $A = 0$ can be approximated, with any given accuracy, by cases when A is small but non-zero. So, without losing generality, we can safely assume that $A \neq 0$. In this case, for $x \stackrel{\text{def}}{=} \theta + k$, where $k \stackrel{\text{def}}{=} \dfrac{B}{A}$, we have

$$\frac{dE}{E} = c \cdot \frac{dx}{x},$$

where $c \stackrel{\text{def}}{=} \dfrac{C}{A}$. Integration leads to $\ln(E) = c \cdot \ln(x) + C_0$ for some constant C_0, thus $E = C_1 \cdot x^c$ for $C_1 \stackrel{\text{def}}{=} \exp(C_0)$, i.e.,

$$E(\theta) = C_1 \cdot (\theta + k)^c. \tag{3.7}$$

Dependence on the bulk stress: conclusion. If we represent $\theta + k$ as $k \cdot \left(\dfrac{\theta}{k} + 1\right)$, then we get the desired dependence of E on θ:

$$E = C_2 \cdot \left(\frac{\theta}{k} + 1\right)^c, \tag{3.8}$$

where $C_2 \stackrel{\text{def}}{=} C_1 \cdot k^c$.

Dependence on the octahedral sheer stress. Similarly, we can conclude that the dependence $E(\tau_{\text{oct}})$ of the elastic modulus E on the octahedral sheer stress τ_{oct} has the form

$$E = C_2' \cdot \left(\frac{\tau_{\text{oct}}}{k'} + 1\right)^{c'}, \tag{3.9}$$

for some constants C_2', k', and c'.

3.4 How to Combine The Formulas Describing Dependence on Each Quantities into a Formula Describing Joint Dependence

Idea. We have used the invariance ideas to derive formulas $E(\theta)$ and $E(\tau_{\text{oct}})$ describing dependence of E on each of the quantities θ and τ_{oct}. Let us now use the same ideas

to combine these two formulas into a single formula describing the dependence on both quantities θ and τ_{oct}.

Based on the previous analysis, for each pair $(\theta, \tau_{\text{oct}})$, we know the value of the modulus E:

- the value $E_1 \stackrel{\text{def}}{=} E(\theta)$ that we obtain if we ignore the octahedral sheer stress and only take into account the bulk stress; and
- the value $E_2 \stackrel{\text{def}}{=} E(\tau_{\text{oct}})$ that we obtain if ignore the bulk stress and only take into account the octahedral sheer stress.

Based on these two values E_1 and E_2, we would like to compute an estimate $E(E_1, E_2)$ for the modulus that would take into account both inputs.

All three values E, E_1, and E_2 represent modulus. Thus, for all three values, only scaling is possible. So, the invariance requirement takes the following form: for every p and q, if we apply the re-scalings $E_1 \to p \cdot E_1$ and $E_2 \to q \cdot E_2$, then the resulting dependence $E(p \cdot E_1, q \cdot E_2)$ has the same form as the original dependence $E(E_1, E_2)$ – after an appropriate re-scaling by some parameter $c(p, q)$ depending on p and q.

So, for every p and every q, there exists a $c(p, q)$ for which, for all E_1 and E_2, we have

$$E(p \cdot E_1, q \cdot E_2) = c(p, q) \cdot E(E_1, E_2). \tag{3.10}$$

Analysis of the problem. If we re-scale only one of the inputs, e.g., E_1, we get

$$E(p \cdot E_1, E_2) = c_1(p) \cdot E(E_1, E_2), \tag{3.11}$$

where $c_1(p) \stackrel{\text{def}}{=} c(p, 1)$. If we first re-scale by p and then by p', then this is equivalent to one re-scaling by $p \cdot p'$. In the first case, we get

$$E((p \cdot p') \cdot E_1, E_2) = E(p' \cdot (p \cdot E_1), E_2) =$$

$$c_1(p') \cdot E(p \cdot E_1, E_2) = c_1(p') \cdot c_1(p) \cdot E(E_1, E_2). \tag{3.12}$$

In the second case, we get

$$E((p \cdot p') \cdot E_1, E_2) = c_1(p \cdot p') \cdot E(E_1, E_2). \tag{3.13}$$

Since the left-hand sides of the equalities (3.12) and (3.13) are equal, their right-hand sides must be equal as well. Dividing the resulting equality by $E(E_1, E_2)$, we conclude that

$$c_1(p \cdot p') = c_1(p) \cdot c_1(p'). \tag{3.14}$$

Differentiating this equality by p' and taking $p' = 1$, we conclude that

$$p \cdot c_1'(p) = c_0 \cdot c_1(p),$$

where $c_0 \stackrel{\text{def}}{=} c_1'(1)$. Thus,

$$\frac{dc_1}{c_1} = c_0 \cdot \frac{dp}{p},$$

so integration leads to $\ln(c_1) = c_0 \cdot \ln(p) + \text{const}$, and

$$c_1(p) = \text{const} \cdot p^{c_0}. \tag{3.15}$$

For $E_1 = 1$, the formula (3.11) takes the form

$$E(p, E_2) = \text{const} \cdot p^{c_0} \cdot E(1, E_2), \tag{3.16}$$

i.e., renaming the variable,

$$E(E_1, E_2) = \text{const} \cdot E_1^{c_0} \cdot E(1, E_2). \tag{3.17}$$

Similarly, we have

$$E(E_1, E_2) = \text{const}' \cdot E_2^{c_0'} \cdot E(E_1, 1), \tag{3.18}$$

for some constants const' and c_0'. In particular, for $E_1 = 1$, the formula (3.18) takes the form

$$E(1, E_2) = \text{const}' \cdot E_2^{c_0'} \cdot E(1, 1). \tag{3.19}$$

Substituting this expression into the formula (3.17), we get

$$E(E_1, E_2) = \text{const} \cdot E_1^{c_0} \cdot \text{const}' \cdot E_2^{c_0'} \cdot E(1, 1). \tag{3.30}$$

Substituting expressions (3.8) and (3.9) for E_1 and E_2 into this formula, we come up with the following conclusion.

Conclusion. From the invariance requirements, we can conclude that the dependence of E on θ and τ_{oct} has the form

$$E(\theta, \tau_{\text{oct}}) = k_1 \cdot \left(\frac{\theta}{k} + 1\right)^{k_2} \cdot \left(\frac{\tau_{\text{oct}}}{k'} + 1\right)^{k_3},$$

where $k_2 = c \cdot c_0$, $k_3 = c' \cdot c_0'$, and

$$k_1 = \text{const} \cdot \text{const}' \cdot E(1,1) \cdot C_2^c \cdot (C_2')^{c'}.$$

Thus, we indeed get a theoretical explanation for the empirical dependence.

Remaining open problems. In this chapter, we used symmetry ideas to provide a solution to one specific physics-related engineering problem: estimating the elastic modulus for unbound aggregate materials.

While there are not too many papers that use symmetries to solve engineering problems, the use of symmetries is ubiquitous in theoretical physics (see, e.g., [16]), and the use of symmetries can help explain many empirical formulas in soft computing [56].

We therefore hope that our example will lead to future application of symmetry ideas in engineering.

Chapter 4

Safety Factors in Soil and Pavement Engineering: Theoretical Explanation of Empirical Data

Comment. The contents of this chapter was published in [24].

4.1 What Is a Safety Factor

Models are approximations to reality. To describe a complex real-life process by a feasible model, we find the most important factors affecting the process and model them. Thus, we ignore small factors; they may be smaller than the factors that we take into account but they still need to be taken into account if we want to provide guaranteed bounds for the desired quantities. To take these small factors into account, engineers multiply the results of the model by a constant known as the *safety factor*.

4.2 Safety Factors in Soil and Pavement Engineering: Empirical Data

In many applications, a safety factor is 2 or smaller. However, in soil and pavement engineering, comparison of the resilient modulus predicted by the corresponding model and the modulus measured by Light Weight Deflectometer shows that, to provide guaranteed

bounds, we need a safety factor of 4; see, e.g., [44]. How can we explain this?

4.3 Explaining the Safety Factor of 2: Reminder

The usual safety factor of 2 is explained, e.g., in [41]; here is the explanation.

Let Δ be the model's estimate. When designing the model, we did not take into account some factors. Let's denote the effect of the largest of these factors by Δ_1. The factors that we ignored are smaller than the one we took into account, so $\Delta_1 < \Delta$, i.e., $\Delta_1 \in [0, \Delta]$. We do not have any reason to assume that any value from the interval $[0, \Delta]$ is more frequent than others; thus, it makes sense to assume that Δ_1 is uniformly distributed on $[0, \Delta]$. Then, the average value of Δ_1 is $\Delta/2$.

The next smallest factor Δ_2 is smaller than Δ_1. The same arguments shows that its average value is $\Delta_1/2$, i.e., $\Delta_2 = 2^{-2} \cdot \Delta$. Similarly, $\Delta_k = 2^{-k} \cdot \Delta$, hence the overall estimate is

$$\Delta + \Delta_1 + \ldots = \Delta + 2^{-1} \cdot \Delta + \ldots + 2^{-k} \cdot \Delta + \ldots = 2\Delta.$$

4.4 A Similar Explanation for the Safety Factor of 4

Empirical data shows that for soil and pavement engineering, 2 is not enough. This means that Δ_1 should be larger than our estimate $\Delta/2$: $\Delta_1 \in [\Delta/2, \Delta]$.

In this case, the average value from this interval is $\Delta_1 = (3/4) \cdot \Delta$.

Similarly, we get $\Delta_2 = (3/4)^2 \cdot \Delta$, $\Delta_k = (3/4)^k \cdot \Delta$ and thus,

$$\Delta + \Delta_1 + \ldots + \Delta_k + \ldots = \Delta \cdot (1 + 3/4 + \ldots + (3/4)^k + \ldots) = \Delta/(1 - 3/4) = 4\Delta.$$

Thus, the safety factor of 4 is indeed explained.

Chapter 5

How Many Monte-Carlo Simulations Are Needed: Case of Interval Uncertainty

In this chapter, we provide a partial answer to the question of how many Monte-Carlo simulations are needed: namely, we provide this answer for the case of interval uncertainty. A recent study of using Monte-Carlo simulations technique for the analysis of different smart electric grid-related algorithms shows that we need approximately 500 simulations to compute the corresponding interval range with 5% accuracy. In this chapter, we provide a theoretical explanation for these empirical results.

Comment. The contents of this chapter was published in [23].

5.1 Formulation of the Problem

Need for interval uncertainty. Data processing means processing measurement results. Measurements are never absolutely accurate: the result $\widetilde{x}$ of measuring a physical quantity is, in general, somewhat different from the actual (unknown) value x of the corresponding quantity.

In the ideal case, we should know which values of the measurement error $\Delta x \stackrel{\text{def}}{=} \widetilde{x} - x$ are possible, and what is the probability of different possible values. These probabilities can be determined if we have a sufficiently large number of situations in which:

- we know the exact values (to be more precise, we have very good estimates of the exact values) and
- we also have measurement results.

In practice, however, we often do not have enough data to determine the corresponding probabilities. In such situations, often, the only information that we have about the measurement error is the upper bound Δ on its absolute value:

$$|\Delta x| \le \Delta;$$

see, e.g., [62]. Then, once we have the measurement result $\widetilde{x}$, the only information that we have about the (unknown) actual value x is that this value belongs to the interval $[\underline{x}, \overline{x}] = [\widetilde{x} - \Delta, \widetilde{x} + \Delta]$. The resulting uncertainty is therefore known as *interval uncertainty*; see, e.g., [32, 43, 54].

Need to propagate interval uncertainty. A data processing algorithm f:

- starts with the results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ of data processing, and
- uses these results to compute an output $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.

This output:

- can be an estimate of some difficult-to-measure quantity, or
- it can be an estimate of the future value of some quantity y.

The corresponding algorithm is usually based on the known relation $y = f(x_1, \ldots, x_n)$ between the actual values of the corresponding quantities. Since, in general, the measurement results $\widetilde{x}_i$ are somewhat different from the actual values x_i, the result $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ of applying the algorithm f to the measurement results is, in general, somewhat different from the actual value $y = f(x_1, \ldots, x_n)$:

$$\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y \ne 0.$$

It is therefore desirable not only to produce the estimate $\widetilde{y}$, but also to find out what the possible values of the corresponding quantity y are.

We know that each quantity x_i can take any values within the corresponding interval $[\underline{x}_i, \overline{x}_i]$. Thus, the desired range of possible values of y have the form

$$\left[\underline{y}, \overline{y}\right] = \{y = f(x_1, \ldots, x_n) : x_1 \in [\underline{x}_1, \overline{x}_1], \ldots, x_n \in [\underline{x}_n, \overline{x}_n]\}.$$

The problem of computing this range is known as the main problem of *interval computations*; see, e.g., [32, 43, 54].

Need for approximate methods. It is known that, in general, the problem of computing the range $\left[\underline{y}, \overline{y}\right]$ exactly is NP-hard; see, e.g., [36]. This means that, unless P=NP (which most computer scientists believe to be impossible), no feasible algorithm can always compute this range exactly. Thus, we need to use approximate methods for computing the desired range.

A natural option: Monte-Carlo technique. One of the natural ways to compute the range is to use *Monte-Carlo* techniques. In this technique, several (N) times:

- we generate random numbers $x_i^{(k)}$ uniformly distributed on the corresponding intervals $[\underline{x}_i, \overline{x}_i]$, and then
- we compute $y^{(k)} = f\left(x_1^{(k)}, \ldots, x_n^{(k)}\right)$.

When $N \to \infty$, the generated random values $x^{(k)} = \left(x_1^{(k)}, \ldots, x_n^{(k)}\right)$, $1 \le k \le N$, cover all parts of the box $[\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_n, \overline{x}_n]$. Thus, in the same limit, the corresponding values $y^{(k)} = f\left(x_1^{(k)}, \ldots, x_n^{(k)}\right)$ fill the entire interval $\left[\underline{y}, \overline{y}\right]$. So, to estimate the desired range, we can use the range formed by the values $y^{(k)}$ corresponding to a sufficiently large number N, namely, the range

$$\left[\min\left(y^{(1)}, \ldots, y^{(N)}\right), \max\left(y^{(1)}, \ldots, y^{(N)}\right)\right].$$

How many simulations do we need? Which value N should we choose? Usually, N is chosen as follows: we repeat the simulations for larger and larger N, and we stop when a further increase in N does not change the resulting range.

Smart electric grid simulations: empirical results. One of the important application areas is the application to electric grids. Electric grids are known to be unstable: a minor change in supply or demand can potentially cause a serious disruption and a blackout. To avoid such situations, engineers employ complicated ("smart") control algorithms.

New improvements for such algorithms are being proposed all the time. To make sure that the new algorithm works well, we need to make sure that the resulting characteristics of the electric grids remain within their stable bounds. Since the parameters of the electric grid are only measured with uncertainty, it is important to make sure that we have stability for all possible combinations of these parameters. One way to do it is to perform Monte-Carlo simulations and to check that the system remains stable for all N resulting combinations $\left(x_1^{(k)}, \ldots, x_n^{(k)}\right)$, $1 \le k \le N$. How many combinations should we choose?

An empirical study [40] showed that if we are interested in 5% accuracy – a typical requirement for data analysis – then we need approximately $N = 500$ simulations to get good results:

- if we have smaller N, e.g., $N = 100$ or $N = 200$, we underestimate the range of y's;
- on the other hand, if we use a larger N – e.g., $N = 1000$ – we do not achieve any significant improvement in comparison to the case $N \approx 500$.

The authors of this study do not have any theoretical explanation for this empirical result.

What we do in this chapter. In this chapter, we provide the desired theoretical explanation.

5.2 Explanation

Accuracy of Monte-Carlo simulations: reminder. It is known (see, e.g., [68]) that if we estimate a quantity based on m measurements, then the relative accuracy of this estimate is

$$\varepsilon \approx \frac{1}{\sqrt{m}}.$$

Let us apply this general feature to our case study. Our goal is to reach the accuracy of $\varepsilon \approx 5\% = 0.05$.

In view of the above formula, to find the number of simulations needed to reach this accuracy, we must find the value m for which

$$\frac{1}{\sqrt{m}} \approx 0.05.$$

This approximate equality is equivalent to

$$\sqrt{m} \approx \frac{1}{0.05} = 20.$$

By squaring both sides of this approximate equality, we get

$$m \approx 20^2 = 400.$$

Taking into account that 500 was not the exact optimal value – it was just better than 100, 200, and 1000 – we conclude that $m = 400$ is a perfect fit for the observed empirical data.

Thus, we provide the desired explanation for the smart electric grid-related simulation results.

Chapter 6

Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation

When learning a dependence from data, to avoid overfitting, it is important to divide the data into the training set and the testing set. We first train our model on the training set, and then we use the data from the testing set to gauge the accuracy of the resulting model. Empirical studies show that the best results are obtained if we use 20-30% of the data for testing, and the remaining 70-80% of the data for training. In this chapter, we provide a possible explanation for this empirical result.

Comment. The contents of this chapter appeared in [25].

6.1 Formulation of the Problem

Training a model: a general problem. In many practical situations, we have a model for a physical phenomenon, a model that includes several unknown parameters. These parameters need to be determined from the known observations; this determination is known as *training* the model.

Need to divide data into training set and testing set. In statistics in general, the more data points we use, the more accurate are the resulting estimates. From this viewpoint, it may seem that the best way to determine the parameters of the model is to

use all the available data points in this determination. This is indeed a good idea if we are absolutely certain that our model adequately describes the corresponding phenomenon.

In practice, however, we are often not absolutely sure that the current model is indeed adequate. In such situations, if we simply use all the available data to determine the parameters of the model, we often get *overfitting* – when the model describes all the data perfectly well without being actually adequate. For example, if we observe some quantity x at n different moments of time, then it is always possible to find a polynomial $f(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + \ldots + a_{n-1} \cdot t^{n-1}$ that will fit all the data points perfectly well — to find such a polynomial, it is sufficient to solve the corresponding system of n linear equations with n unknowns $a_0, \ldots, a_{n-1}$:

$$a_0 + a_1 \cdot t_1 + a_2 \cdot t_i^2 + \ldots + a_{n-1} \cdot t_i^{n-1}, \quad i = 1, \ldots, n.$$

This does not mean that the resulting model is adequate, i.e., that the resulting polynomial can be used to predict the values $x(t)$ for all t: one can easily show that if we start with noisy data, the resulting polynomial will be very different from the actual values of $x(t)$. For example, if $n = 1$ and the actual value of $x(t)$ is a constant, then, due to noise, the resulting polynomial $x(t) = a_0 + a_1 \cdot t$ will be a linear function with $a_1 \neq 0$. Thus, for large t, we will have $x(t) \to \infty$, so the predicted values will be very different from the actual (constant) value of the signal.

To avoid overfitting, it is recommended that we divide the observations into training and testing data:

- First, we use the training data to determine the parameters of the model.
- After that, we compare the model's predictions for all the testing data points with what we actually observed, and use this comparison to gauge the accuracy of our model.

Which proportion of data should we allocate for testing? Empirical analysis has shown that the best results are attained if we allocate 20-30% of the original data points

for testing, and use the remaining 70-80% for training.

For this division, we get accuracy estimates which are:

- valid – in the sense that they do not overestimate the accuracy (i.e., do not underestimate the approximation error), and
- are the more accurate among the valid estimates – i.e., their overestimation of the approximation error is the smallest possible.

What we do in this chapter. In this chapter, we provide a possible explanation for this empirical fact.

6.2 Formal Description and Analysis of the Problem

Training and testing: towards a formal description. Our goal is to find the dependence of the desired quantity y on the corresponding inputs $x_1, \ldots, x_n$. To be more specific, we assume that the dependence has the form

$$y = f(a_1, \ldots, a_m, x_1, \ldots, x_n),$$

for some parameters $a_1, \ldots, a_m$. For example, we can assume that the dependence is linear, in which case $m = n + 1$ and

$$y = a_1 \cdot x_1 + \ldots + a_n \cdot x_n + a_{m+1}.$$

We can assume that the dependence is quadratic, or sinusoidal, etc.

To find this dependence, we use the available data, i.e., i.e., we use N situations $k = 1, \ldots, n$ in each of which we know both the values of the inputs $x_1^{(k)}, \ldots, x_n^{(k)}$ and the corresponding output $y^{(k)}$.

Let p denote the fraction of the data that goes into the training set. This means that out of the original N patterns $\left(x_1^{(k)}, \ldots, x_n^{(k)}, y^{(k)}\right)$:

- $N \cdot p$ patterns form a training set, and
- the remaining $(1-p) \cdot N$ patterns form a testing set.

We use the training set to find estimates $\widehat{a}_1, \ldots, \widehat{a}_m$ of the parameters $a_1, \ldots, a_m$. Then, for each pattern $\left(x_1^{(k)}, \ldots, x_n^{(k)}, y^{(k)}\right)$ from the testing set, we compare the desired output $y^{(k)}$ with the result

$$\widehat{y}^{(k)} = f\left(\widehat{a}_1, \ldots, \widehat{a}_m, x_1^{(k)}, \ldots, x_n^{(k)}\right)$$

of applying the trained model to the inputs. Based on the differences

$$d_k \stackrel{\text{def}}{=} y^{(k)} - \widehat{y}^{(k)},$$

we gauge the accuracy of the trained model.

How do we gauge the accuracy of the model. Many different factors influence the fact that the resulting model is not perfect, such as measurement errors, approximate character of the model itself, etc.

It is known that under reasonable assumptions, the distribution of a joint effect of many independent factors s close to Gaussian (normal) – the corresponding mathematical result is known as the *Central Limit Theorem*; see, e.g., [68]. Thus, we can safely assume that the differences d_k are normally distributed.

It is known that a 1-D normal distribution is uniquely determined by two parameters: mean value μ and standard deviation σ. Thus, based on the differences d_k, we can estimate:

- the mean value (bias) of the trained model, and
- the standard deviation σ describing the accuracy of the trained model.

A general fact from statistics: reminder. In statistics, it is known that when we use M values to estimate a parameter, the standard deviation of the estimate decreases by a factor of $\sqrt{M}$.

Example. The factor-of-$\sqrt{M}$ decrease is the easiest to explain on the simplest example when have a single quantity q, and we perform several measurements of this quantity by using a measuring instrument for which the standard deviation of the measurement error is σ_0. As a result, we get M measurement results $q_1, \ldots, q_M$. As an estimate for q, it is reasonable to take the arithmetic mean

$$\widehat{q} = \frac{q_1 + \ldots + q_M}{M}.$$

Then, the resulting estimation error $\widehat{q} - q$, i.e., the difference between this estimate and the actual (unknown) value q of the quantity of interest has the form

$$\widehat{q} - q = \frac{q_1 + \ldots + q_M}{M} - q = \frac{(q_1 - q) + \ldots + (q_M - q)}{M}.$$

By definition, for each difference $q_i - q$, the standard deviation is equal to σ_0. and thus, the variance is equal to σ_0^2.

Measurement errors corresponding to different measurements are usually independent. It is known that the variance of the sum of independent random variables is equal to the sum of the variances. Thus, the variance of the sum $(q_1 - q) + \ldots + (q_M - q)$ is equal to $M \cdot \sigma_0^2$, and the corresponding standard deviation is equal to $\sqrt{M \cdot \sigma_0^2} = \sqrt{M} \cdot \sigma_0$. When we divide the sum by M, the standard deviation also divides by the same factor. So, the standard deviation of the difference $\widehat{q} - q$ is equal to $\dfrac{\sqrt{M} \cdot \sigma_0}{M} = \dfrac{\sigma_0}{\sqrt{M}}$.

Let us use the general fact from statistics. We estimate the parameters of the model based on the training set, with $p \cdot N$ elements. Thus, the standard deviation of the corresponding model is proportional to $\dfrac{1}{\sqrt{p \cdot N}}$.

When we gauge the accuracy of the model, we compare the trained model with the data from the testing set. Even if the trained model was exact, because of the measurement errors, we would not get the exact match. Instead, based on $(1-p) \cdot N$ measurements, we would get the standard deviation proportional to $\dfrac{1}{\sqrt{(1-p) \cdot N}}$.

We want to estimate the difference d_k between the trained model and the testing data. It is reasonable to assume that, in general, the errors corresponding to the training set

and to the testing set are independent – we may get positive correlation in some cases, negative correlation in others, so, on average, the correlation is 0. For independence random variables, the variance is equal to the sum of the variances. Thus, on average, this variance is proportional to

$$\left(\frac{1}{\sqrt{p \cdot N}}\right)^2 + \left(\frac{1}{\sqrt{(1-p) \cdot N}}\right)^2 = \frac{1}{p \cdot N} + \frac{1}{(1-p) \cdot N} = \frac{1}{(p \cdot (1-p)) \cdot N}.$$

Thus, to get the smallest possible estimate for the approximation error, then, out of all possible values p, we need to select the value p for which the product $p \cdot (1-p)$ is the largest possible.

Which values p are possible? The only remaining question is now: which values p are possible?

Our requirement was that we should select p for which the gauged accuracy is guaranteed not to overestimate the accuracy. In precise terms, this means that the standard deviation of the trained model – i.e., the standard deviation of the estimate $\widehat{y}^{(k)}$ – should be smaller than or equal to the standard deviation of the difference d_k by which we gauge the model's accuracy:

$$\sigma\left[\widehat{y}^{(k)}\right] \le \sigma[d_k].$$

Here, $d_k = \widehat{y}^{(k)} - y^{(k)}$ is the difference between:

- the estimate $\widehat{y}^{(k)}$ whose inaccuracy is cased by the measurement errors of the training set and
- the value $y^{(k)}$ whose inaccuracy is cased by the measurement errors of the testing set.

So, we must have

$$\sigma\left[\widehat{y}^{(k)}\right] \le \sigma\left[\widehat{y}^{(k)} - y^{(k)}\right].$$

In general, for two random variables r_1 and r_2 with standard deviations $\sigma[r_1]$ and $\sigma[r_2]$, the smallest possible value of the standard deviation of the difference is $|\sigma[r_1 -]\sigma[r_2]|$ (see, e.g., [68]):

$$\sigma[r_1 - r_2] \ge |\sigma[r_1] - \sigma[r_2]|.$$

In particular, for the difference $d_k = \widehat{y}^{(k)} - y^{(k)}$, the smallest possible value of its standard deviation $\sigma\left[\widehat{y}^{(k)} - y^{(k)}\right]$ is

$$\left|\sigma\left[\widehat{y}^{(k)}\right] - \sigma\left[y^{(k)}\right]\right|.$$

Thus, to make sure that we do not underestimate the measurement error, we must guarantee that

$$\sigma\left[\widehat{y}^{(k)}\right] \le \left|\sigma\left[\widehat{y}^{(k)}\right] - \sigma\left[y^{(k)}\right]\right|,$$

i.e., that $a \le |a - b|$, where we denoted $a \stackrel{\text{def}}{=} \sigma\left[\widehat{y}^{(k)}\right]$ and $b \stackrel{\text{def}}{=} \sigma\left[y^{(k)}\right]$.

In principle, we can have two different cases: $a \le b$ and $b \le a$. Let us consider these two cases one by one.

- If $a \ge b$, then the desired inequality takes the form $a \le a - b$, which for $b > 0$ is impossible.
- Thus, we must have $b \le a$. In this case, the above inequality takes the form $a \le b - a$, i.e., equivalently, $2a \le b$.

Thus, we must have

$$2\sigma\left[\widehat{y}^{(k)}\right] \le \sigma\left[y^{(k)}\right].$$

Since the inaccuracy of the estimate $\widehat{y}^{(k)}$ comes only from measurement errors of the training set, with $p \cdot N$ elements, we have

$$\sigma\left[\widehat{y}^{(k)}\right] = \frac{\sigma_0}{\sqrt{p \cdot N}}$$

for some σ_0. Similarly, since the inaccuracy of the estimate $y^{(k)}$ comes only from measurement errors of the testing set, with $(1-p) \cdot N$ elements, we have

$$\sigma\left[y^{(k)}\right] = \frac{\sigma_0}{\sqrt{(1-p) \cdot N}}.$$

Thus, the above inequality takes the form

$$2 \cdot \frac{\sigma_0}{\sqrt{p \cdot N}} \le \frac{\sigma_0}{\sqrt{(1-p) \cdot N}}.$$

Dividing both sides of this inequality by σ_0 and multiplying by $\sqrt{N}$, we conclude that

$$\frac{2}{\sqrt{p}} \le \frac{1}{\sqrt{1-p}}.$$

Squaring both sides, we get

$$\frac{4}{p} \le \frac{1}{1-p}.$$

By bringing both sides to the common denomination, we get $4-4p \le p$, i.e., $4 \le 4p+p = 5p$ and $p \ge 0.8$.

Thus, to make sure that our estimates do not overestimate accuracy, we need to select the values $p \ge 0.8$.

Towards the final conclusion. As we have mentioned earlier, out of all possible values p, we need to select a pone for which the product $p \cdot (1-p)$ is the largest possible. For $p \ge 0.8$, the function $p \cdot (1-p)$ is decreasing. Thus, its largest values is attained when the value p is the smallest possible – i.e., when $p = 0.8$.

So, we have indeed explained why $p \approx 80\%$ is empirically the best division into the training and the testing sets.

Chapter 7

How to Use Absolute-Error-Minimizing Software to Minimize Relative Error: Practitioner's Guide

In many engineering and scientific problems, there is a need to find the parameters of a dependence from the experimental data. There exist several software packages that find the values for these parameters – values for which the mean square value of the absolute approximation error is the smallest. In practice, however, we are often interested in minimizing the mean square value of the *relative* approximation error. In this chapter, we show how we can use the absolute-error-minimizing software to minimize the relative error.

Comment. The contents of this chapter was published in [21].

7.1 Formulation of the Problem

Practical problem. In many practical situations, we know that the dependence between the quantity y and related quantities $x_1, \ldots, x_n$ has the known form, i.e., the form $y = f(x_1, \ldots, x_n, c_1, \ldots, c_m)$ with a known function f and unknown values of the coefficients $c_1, \ldots, c_m$.

Examples. For example, we may know that y linearly depends on x_i, in which case the

desired dependence has the form

$$y = c_1 \cdot x_1 + \ldots + c_n \cdot x_n + c_{m+1}$$

for some unknown values c_i.

We may have a quadratic dependence, in which case

$$y = c_1 \cdot x_1 + \ldots + c_n \cdot y_n + c_{n+1} + c_{n+2} \cdot x_1^2 + c_{n+3} \cdot x_1 \cdot x_2 + \ldots$$

These two cases are particular cases of a more general situation in which the dependence of the expression $f(x_1, \ldots, x_n, c_1, \ldots, c_m)$ on the coefficients c_j is linear, i.e., in which

$$f(x_1, \ldots, x_n, c_1, \ldots, c_m) = \sum_{j=1}^{m} c_j \cdot f_j(x_1, \ldots, x_n)$$

for given functions $f_1(x_1, \ldots, x_n)$, $\ldots$, $f_m(x_1, \ldots, x_n)$.

We may also have a more complex dependence, in which the dependence on the coefficients c_j is nonlinear. For example, in radioactive decay, once we know the initial amount y_0 of the radioactive material, then the amount y remaining after time x_1 is described by the formula $y = y_0 \cdot e^{-c_1 \cdot x_1}$, for an appropriate to-be-determined coefficient c_1.

What information we have to solve this problem. In the above situation, it is necessary to determine the values of these parameters $c_1, \ldots, c_m$ from the experimental data.

In engineering, geosciences, and in many other application areas, this is known as the *inverse problem* – as opposed to the *forward problem*, when we know the values of x_i and c_j, and we use the dependence $y = f(x_1, \ldots, x_n, c_1, \ldots, c_m)$ to predict the value of the quantity y.

To solve the inverse problem, we can use the results coming out of several (K) situations in which both y and x_i have been measured, and we use the results $y^{(k)}$ and $x_i^{(k)}$ of these measurements to find the coefficients c_i for which, for each k from 1 to K, we have

$$y^{(k)} \approx f\left(x_1^{(k)}, \ldots, x_n^{(k)}, c_1, \ldots, c_m\right).$$

It is important to emphasize that since measurements are never absolutely accurate – there is always some measurement error – the measurement result $y^{(k)}$ is only approximately equal to the quantity $f\left(x_1^{(k)}, \ldots, x_n^{(k)}, c_1, \ldots, c_m\right)$; see, e.g., [62].

There exist many software packages that minimize the absolute error. The standard approach to solving the above problem is the Least Squares approach (see, e.g., [68]), in which we find the coefficients $c_1, \ldots, c_m$ for which the mean square value of the (absolute) approximation error is minimized, i.e., that minimize the expression $\sum\limits_{k=1}^{K} \left(\Delta y^{(k)}\right)^2$, where by $\Delta y^{(k}$, we denoted the (absolute) approximation errors

$$\Delta y^{(k)} \stackrel{\text{def}}{=} y^{(k)} - f\left(x_1^{(k)}, \ldots, x_n^{(k)}, c_1, \ldots, c_m\right).$$

For example, MatLab has such programs – both for the linear and for the nonlinear cases. In addition, neural network packages – in particular, the MatLab neural network toolbox – minimize the above sum of squares; see, e.g., [9].

Some of these packages only deal with situations in which the dependence $f\left(x_1, \ldots, x_n, c_1, \ldots, c_m\right)$ is linear in terms of the coefficients. In these cases, the above problem takes the form

$$y^{(k)} \approx \sum_{j=1}^{m} c_j \cdot f_j\left(x^{(1)}, \ldots, x_n^{(k)}\right),$$

and the Least Squares idea means minimizing the sum $\sum\limits_{j=1}^{m} \left(\Delta y^{(k)}\right)^2$, where

$$\Delta y^{(k)} = y^{(k)} - \sum_{j=1}^{m} c_j \cdot f_j\left(x^{(1)}, \ldots, x_n^{(k)}\right).$$

These packages usually assume that we know the values $q^{(k)}$ and $p_1^{(k)}, \ldots, p_m^{(k)}$ for all k from 1 to K, and they find the coefficients $c_1, \ldots, c_m$ for which the sum

$$\sum_{k=1}^{K} \left(q^{(k)} - \sum_{j=1}^{m} c_j \cdot p_j^{(k)}\right)^2$$

attains its smallest possible value. To apply such a package to our problem, it is sufficient to take $q^{(k)} = y^{(k)}$ and $p_j^{(k)} = f_j\left(x_1^{(k)}, \ldots, x_n^{(k)}\right)$.

Often, we need to minimize relative errors instead. While in many practical situations, minimizing the absolute approximation errors is a reasonable idea, in many other situations, it is more appropriate to minimize *relative* approximation errors

$$\delta y^{(k)} \stackrel{\text{def}}{=} \frac{\Delta y^{(k)}}{y^{(k)}},$$

i.e., to be precise, to minimize the sum $\sum\limits_{k=1}^{K} \left(\delta y^{(k)}\right)^2$.

For example, if the desired dependence covers a wide range of possible values of y, e.g., ranging from $y = 10$ to $y = 1000$, it makes more sense to want to approximate all these values with the same relative accuracy – e.g., 5% or 10% – than with the same absolute accuracy, say, 5 – in the case of absolute accuracy, we have a very crude 50% accurate approximation for small values $y \approx 10$ and an unnecessarily accurate (0.5% accurate) approximation of values $y \approx 1000$ (unnecessarily accurate since we usually cannot even measure y with such a high accuracy).

Problem. The problem is that, in contrast to minimization of absolute error, for which there are several available software packages, not many packages are available for minimizing the relative error.

In principle, we can write our own code for solving this problem, but it would be much easier if we could simply use the existing software.

What we do in this chapter. In this chapter, we show how we can use absolute-error-minimizing software to minimize relative errors. In Section 2, we describe how to do it for the case when the dependence on the coefficients c_j is linear. In Section 3, we consider the general case, when the dependence on the coefficients c_j may be non-linear.

7.2 Case When the Dependence on the Coefficients c_j is Linear

Description of the case: reminder. We assume that

$$y = \sum_{j=1}^{m} c_j \cdot f_j\left(x_1, \ldots, x_n\right),$$

for known functions $f_1\left(x_1, \ldots, x_n\right)$, $\ldots$, $f_m\left(x_1, \ldots, x_m\right)$. Based on the results $y^{(k)}$ and $x_i^{(k)}$ of measuring the corresponding quantities, we want to find the coefficients $c_1, \ldots, c_m$ for which

$$y^{(k)} \approx \sum_{j=1}^{m} c_j \cdot f_j\left(x_1^{(k)}, \ldots, x_n^{(k)}\right)$$

for all k from 1 to K. To be more precise, we want to find the values $c_1, \ldots, c_m$ for which the sum $\sum\limits_{k=1}^{K} \left(\delta y^{(k)}\right)^2$ is the smallest possible, where $\delta y^{(k)} \stackrel{\text{def}}{=} \dfrac{\Delta y^{(k)}}{y^{(k)}}$ and

$$\Delta y^{(k)} \stackrel{\text{def}}{=} y^{(k)} - \sum_{j=1}^{m} c_j \cdot f_j\left(x_1^{(k)}, \ldots, x_n^{(k)}\right).$$

We would like to use a Least Squares package. To solve our problem, we would like to use a Least Squares package, that, given the values $q^{(k)}$ and $p_1^{(k)}, \ldots, p_m^{(k)}$ for all k from 1 to K, find the values c_j that minimize the sum

$$\sum_{k=1}^{K} \left(p^{(k)} - \sum_{j=1}^{m} c_j \cdot p_j^{(k)}\right)^2.$$

Analysis of the problem. Substituting the expression

$$\Delta y^{(k)} = y^{(k)} - \sum_{j=1}^{m} c_j \cdot f_j\left(x_1^{(k)}, \ldots, x_m^{(k)}\right)$$

into the definition of the relative approximation error $\delta y^{(k)}$, we conclude that

$$\delta y^{(k)} = 1 - \sum_{j=1}^{m} c_j \cdot \frac{f_j\left(x_1^{(k)}, \ldots, x_n^{(k)}\right)}{y^{(k)}}.$$

Thus, the problem of minimizing the sum $\sum\limits_{k=1}^{K}\left(\delta y^{(k)}\right)^2$ of the squares of relative errors is equivalent to the problem of minimizing the sum

$$\sum_{k=1}^{K}\left(q^{(k)}-\sum_{j=1}^{m}c_j\cdot p_j^{(k)}\right)^2,$$

with $q^{(k)}=1$ and $p_j^{(k)}=\dfrac{f_j\left(x_1^{(k)},\ldots,x_n^{(k)}\right)}{y^{(k)}}$. So, we arrive at the following recommendation.

Recommendation. Our recommendation is to find the coefficients $c_1,\ldots,c_m$ by applying a Least Squares package to the values $q^{(k)}=1$ and

$$p_j^{(k)}=\frac{f_j\left(x_1^{(k)},\ldots,x_n^{(k)}\right)}{y^{(k)}}.$$

7.3 General (Possibly Nonlinear) Case

First idea: description. The above approach can be naturally extended to the nonlinear case. Namely, minimizing the sum of relative errors

$$\sum_{k=1}^{K}\left(\delta y^{(k)}\right)^2=\sum_{k=1}^{K}\left(\frac{y^{(k)}-f\left(x_1^{(1)},\ldots,x_n^{(k)},c_1,\ldots,c_m\right)}{y^{(k)}}\right)^2$$

is equivalent to minimizing the sum of the absolute differences

$$\sum_{k=1}^{K}\left(z^{(k)}-g\left(x_1^{(1)},\ldots,x_n^{(k)},y^{(k)},c_1,\ldots,c_m\right)\right)^2,$$

where $z^{(k)}=1$ and

$$g\left(x_1,\ldots,x_n,y,c_1,\ldots,c_m\right)\stackrel{\text{def}}{=}\frac{f\left(x_1,\ldots,x_n,c_1,\ldots,c_m\right)}{y}.$$

Thus, we arrive at the following recommendation.

First idea: resulting recommendations. To minimize the relative error, apply the absolute-error-minimizing software to find the coefficients $c_1, \ldots, c_m$ from the condition that

$$z^{(k)} \approx g\left(x_1^{(1)}, \ldots, x_n^{(k)}, y^{(k)}, c_1, \ldots, c_m\right)$$

for $k = 1, \ldots, K$, where we denoted $z^{(k)} = 1$ and

$$g\left(x_1, \ldots, x_n, y, c_1, \ldots, c_m\right) \stackrel{\text{def}}{=} \frac{f\left(x_1, \ldots, x_n, c_1, \ldots, c_m\right)}{y}.$$

Second idea: description. Alternatively, we can apply the existing software to approximate $\ln(y)$ by the dependence $\ln\left(f\left(x_1, \ldots, x_n, c_1, \ldots, x_m\right)\right)$, i.e., find the coefficients $c_1, \ldots, c_m$ for which the sum $\sum\limits_{k=1}^{K}\left(\Delta Y^{(k)}\right)^2$ is the smallest possible, where $\Delta Y^{(k)} \stackrel{\text{def}}{=} Y^{(k)} - F\left(x_1^{(k)}, \ldots, x_n^{(k)}, c_1, \ldots, c_m\right)$, $Y^{(k)} \stackrel{\text{def}}{=} \ln\left(y^{(k)}\right)$ and

$$F\left(x_1, \ldots, x_n, c_1, \ldots, c_m\right) \stackrel{\text{def}}{=} \ln\left(f\left(x_1, \ldots, x_n, c_1, \ldots, c_m\right)\right).$$

Second idea: justification. Measurement errors are usually relatively small, so we can safely ignore terms which are quadratic (or of higher order) in terms of the corresponding error. For example, if we measure with accuracy 10%, then the square of the corresponding measurement error is about 1%, which is much smaller than 10% and can, therefore, be safely ignored.

In our case, by definition of the approximation error $\Delta y^{(k)}$, we have

$$y^{(k)} = f\left(x_1^{(k)}, \ldots, x_n^{(k)}, c_1, \ldots, c_m\right) + \Delta y^{(k)}.$$

So,

$$\ln\left(y^{(k)}\right) = \ln\left(f\left(x_1^{(k)}, \ldots, x_n^{(k)}, c_1, \ldots, c_m\right) + \Delta y^{(k)}\right).$$

If we expand this expression in Taylor series in terms of $\Delta y^{(k)}$ and take into account that the derivative of logarithm $\ln(x)$ is $\dfrac{1}{x}$, we conclude that

$$\ln\left(y^{(k)}\right) = \ln\left(f\left(x_1^{(k)}, \ldots, x_n^{(k)}, c_1, \ldots, c_m\right)\right) +$$

$$\frac{1}{f\left(x_1^{(k)},\ldots,x_n^{(k)},c_1,\ldots,c_m\right)}\cdot\Delta y^{(k)}+\ldots$$

As we have mentioned, in this expansion, we can safely ignore terms which are quadratic (or of higher order) in $\Delta y^{(k)}$, thus

$$\ln\left(y^{(k)}\right)=\ln\left(f\left(x_1^{(k)},\ldots,x_n^{(k)},c_1,\ldots,c_m\right)\right)+$$

$$\frac{1}{f\left(x_1^{(k)},\ldots,x_n^{(k)},c_1,\ldots,c_m\right)}\cdot\Delta y^{(k)}.$$

The expression in the denominator is equal to $y^{(k)}-\Delta y^{(k)}$, thus

$$\ln\left(y^{(k)}\right)=\ln\left(f\left(x_1^{(k)},\ldots,x_n^{(k)},c_1,\ldots,c_m\right)\right)+\frac{1}{y^{(k)}-\Delta y^{(k)}}\cdot\Delta y^{(k)}.$$

Expanding again in terms of $\Delta y^{(k)}$ and ignoring quadratic and higher order terms, we conclude that

$$\ln\left(y^{(k)}\right)=\ln\left(f\left(x_1^{(k)},\ldots,x_n^{(k)},c_1,\ldots,c_m\right)\right)+\frac{\Delta y^{(k)}}{y^{(k)}},$$

i.e., by definition of the relative error $\delta y^{(k)}$, that

$$\ln\left(y^{(k)}\right)-\ln\left(f\left(x_1^{(k)},\ldots,x_n^{(k)},c_1,\ldots,c_m\right)\right)=\delta y^{(k)}.$$

Thus,

$$\Delta Y^{(k)}=Y^{(k)}-F\left(x_1^{(1)},\ldots,x_n^{(k)},c_1,\ldots,c_m\right)=$$

$$\ln\left(y^{(k)}\right)-\ln\left(f\left(x_1^{(k)},\ldots,x_n^{(k)},c_1,\ldots,c_m\right)\right)=\delta y^{(k)},$$

i.e., $\Delta Y^{(k)}=\delta y^{(k)}$.

So, minimizing the sum $\sum\limits_{k=1}^{K}\left(\Delta Y^{(k)}\right)^2$ is indeed equivalent to the minimization of the mean squares value of the relative error, i.e., to the minimization of the sum $\sum\limits_{k=1}^{K}\left(\delta y^{(k)}\right)^2$. Hence, we arrive at the following recommendation.

Second idea: recommendation. To solve the desired relative-error-minimization problem, we compute the values $Y^{(k)}\stackrel{\text{def}}{=}\ln\left(y^{(k)}\right)$ and form a new function $F\left(x_1,\ldots,x_n,c_1,\ldots,c_m\right)=\ln\left(f\left(x_1,\ldots,c_1,\ldots,c_m\right)\right)$.

Then, we apply the absolute-error-minimizing package to find the coefficients $c_1, \ldots, c_m$ for which

$$Y^{(k)} \approx F\left(x_1^{(k)}, \ldots, x_n^{(k)}, c_1, \ldots, c_m\right)$$

for all $k = 1, \ldots, K$.

Chapter 8

How to Best Apply Deep Neural Networks in Geosciences: Towards Optimal "Averaging" in Dropout Training

The main objectives of geosciences is to find the current state of the Earth – i.e., solve the corresponding *inverse problems* – and to use this knowledge for predicting the future events, such as earthquakes and volcanic eruptions. In both inverse and prediction problems, often, machine learning techniques are very efficient, and at present, the most efficient machine learning technique is deep neural training. To speed up this training, the current deep learning algorithms use *dropout* techniques: they train several sub-networks on different portions of data, and then "average" the results. A natural idea is to use arithmetic mean for this "averaging", but empirically, geometric mean works much better. In this chapter, we provide a theoretical explanation for the empirical efficiency of selecting geometric mean as the "averaging" in dropout training.

Comment. The contents of this chapter was published in [26].

8.1 Introduction

Main objectives of science. The main objectives of science are:

- to determine the state of the world, and
- based on this knowledge, to predict the future state of the world.

For example, in geosciences:

- we want to determine the density at different depths and at different locations based on the observed data – seismic, gravitational, etc. (this is known as the *inverse problem*) and
- based on this knowledge, we would like to be able to predict catastrophic events such as earthquake and volcanic eruptions (this is known as the *prediction problem*).

Machine learning is often needed. In some situations, we know the equations describing the physical phenomena, and we can use these equations to make necessarily determinations and predictions. This is how, e.g., weather is predicted.

In many other situations, however, we either do not know the exact equations – or these equations are too difficult to solve. In such situations, instead of using specific equations, we can use general *machine learning* tools. In both problems:

- we start with a tuple of measurement results x, and
- we would like to estimate the tuple y of the desired quantities – e.g., the density values or the values describing the future volcanic activity.

To make this prediction, we need to have a database of *patterns*, i.e., pairs $\left(x^{(k)}, y^{(k)}\right)$ corresponding to past situations in which we know both x and y.

For example, if we are interested in predicting volcanic activity at least a week in advance, we need to use patterns in which $y^{(k)}$ is the observed volcanic activity and $x^{(k)}$ are measurement results performed at least a week before the corresponding activity.

Which machine learning techniques should we use: need for deep learning. Learning is what living creatures have to do in order to survive. To learn, living creatures

use signals processed a network of special cells – neurons. It is reasonable to assume that as a result of billions of years of improving winner-takes-all evolution, nature has come up with an optimal – or near-optimal – way of learning. And indeed, artificial neural networks – that are based on simulating networks of biological neurons – are, at present, the most efficient machine learning technique; see, e.g., [27].

Specifically, the most efficient technique involves *deep learning*, where we have a large number of layers with reasonably few neurons in each layer; the advantages of such an arrangement are presented in [5, 27].

Deep neural networks has indeed been efficient in geosciences, both in inverse problem (see, e.g., [20]) and in prediction problem (see, e.g., [19, 58, 59, 60]).

Need to speed up the learning process. To get a good description of the corresponding phenomenon, it is desirable to have a large number of patterns. As a result, training on all these patterns takes time. It is thus desirable to speed up computations.

When a person has a task that takes too long to do it by him/herself, a natural idea of speeding it up is to ask for help and to have several people performing this task in parallel. Similarly, a natural way to speed up computations is to perform them in parallel, on several processors.

Need to speed up the learning process naturally leads to dropout training. For traditional neural networks, when we had a large number of neurons in each layer, parallelization was reasonably natural: we just divide the neurons into several groups, and have each processor simulate neurons from the corresponding group.

However, for deep neural networks, there is a relatively small number of neurons in each layer, so we cannot apply the above natural parallelization. A natural alternative idea is:

- to use different parts of the data for training (in parallel) on different sub-networks of the network, and
- then to "average" the results.

Since for each of these trainings, we drop some of the patterns and some of the neurons, this idea is known as a *dropout*; see, e.g., [27, 69, 70].

Which "averaging" works better in deep learning-related dropout training? What is the best way to "average" the values $v_1, \ldots, v_m$ obtained from different parallel trainings? The original idea was to use an arithmetic average, i.e., to use the value v for which

- adding m identical copies of the value v leads to exactly the same result as
- adding m training results $v_1, \ldots, v_m$:

$$v_1 + \ldots + v_m = v + \ldots + v.$$

In this scheme, we get

$$v = \frac{v_1 + \ldots + v_m}{m}.$$

However, it turned out that better results are attained if, instead of addition, we use different combination rules $a * b$. In this case, as the result of such "averaging", we take the value v for which

$$v_1 * \ldots * v_m = v * \ldots * v.$$

In particular, it turned out the empirically, the best results are attained if, as a combination $a * b$, we use product instead of the sum [27, 75]. In this case, the result of "averaging" is the geometric mean:

$$v = \sqrt[m]{v_1 \cdot \ldots \cdot v_m}.$$

Comment. Usually, the values are re-scaled, so that they fit into an interval, e.g., $[0, 1]$. So, without losing generality, we can assume that all the averaged values v_i are non-negative.

How can we explain this empirical success? The paper [75] has some *qualitative* explanations of why geometric mean works better than arithmetic one in deep-learning

related dropout training. However, it does not provide a *quantitative* explanation of why namely the "averaging" based on multiplication works best.

What we do in this chapter. In this chapter, we provide an explanation for the empirical success of geometric mean in deep learning-related dropout training. To be more precise, we list all "averaging" operations corresponding to optimal combination functions – optimal under all possible reasonable optimality criteria. As a result, we get a 1-D family of possible "averaging" operations – and it turns out that this list contains arithmetic and geometric means as particular cases.

Thus, we provide a quantitative explanation of the empirical success of geometric mean in deep learning-related dropout training.

Comment. Cannot we do better and explain why *only* the geometric mean is the best? Probably this is possible if we explicitly select *one* optimality criterion. However, in our general formulation, when we allow *all* possible optimality criteria, the appearance of the arithmetic average is inevitable: it corresponds, for example, to using the Least Squares optimality criterion

$$\sum_{i=1}^{m}(v_i - v)^2 \to \min,$$

a criterion that often makes sense in machine learning; see, e.g., [9, 27].

8.2 What Is a Combination Operation? What Is a Reasonable Optimality Criterion? Towards Precise Definitions

What is a combination operation? A combination operation (also known as an *aggregation operation* or an *aggregation function*) $a * b$ is a function that maps two non-negative numbers a and b into a non-negative number $a * b$.

There are many different combination operations; see, e.g., [7, 10, 12, 13, 28, 71]. What

are the reasonable properties of the combination operations used in deep learning-related dropout training?

First reasonable property of a combination operation used in deep learning-related dropout training: commutativity. We have several results v_i that were obtained by using the same methodology – the only difference is that we randomly selected a different set patterns and we randomly selected a different sub-network. From this viewpoint, there is no reason to believe that some of these results are more valuable than others. Thus, it makes sense to require that the result of combining two values should not depend on the order in which they are presented, i.e., that $a * b = b * a$ for all a and b.

In other words, it is reasonable to require that the combination operation be commutative.

Second reasonable property of a combination operation used in deep learning-related dropout training: associativity. If we have three values a, b, and c, then we can:

- first combine a and b and get $a * b$, and
- add, combine the result $a * b$ with c, resulting in $(a * b) * c$.

Alternatively, we can:

- first combine b nd c into a single value $b * c$, and
- then combine a with the result $b * c$ of combining b and c, thus getting

$$a * (b * c).$$

It is reasonable to require that the result of combining the three values should not depend on the order in which we combine them, i.e., that we should have

$$(a * b) * c = a * (b * c).$$

In other words, it is reasonable to require that the combination operation be associative.

Third reasonable property of a combination operation used in deep learning-related dropout training: monotonicity. It is reasonable to require that if one of the combined values increases, then the result of the combination should also increase (or at least not decrease). In other words, it is reasonable to require that $a * b$ is a (non-strictly) increasing function of each of the variables:

- if $a \le a'$, then $a * b \le a' * b$, and
- if $b \le b'$, the $a * b \le a * b'$.

Final reasonable property of a combination operation used in deep learning-related dropout training: continuity. In practice, all the values are estimated only approximately. It is therefore reasonable to require that a small difference between the ideal value v_i and the corresponding approximate computational result should not drastically affect the result of the combination.

In precise terms, this means that the operation $a * b$ should be continuous.

Towards the resulting definition of a combination function. So, we define a combination operation as a commutative, associative, monotonic continuous function $a * b$ of two real non-negative variables.

What is optimality criterion: a general discussion. Out of all possible combination operations $*$, we should select the one which is, in some reasonable sense, optimal for deep learning-related dropout training. How can we describe the corresponding optimality criterion?

In many practical problems, when we talk about optimization, we have an objective function whose value we want to maximize or minimize. However, this is not the most general case of optimization.

For example, if we select an algorithm a for solving a certain problem, and we are interested in achieving the fastest possible average computation time $A(a)$, we may end up with several different algorithms a, a', ..., that have the exact same average computation time $A(a) = A(a') = \ldots$ In this case, it makes sense to use this non-uniqueness to optimize something else: e.g., the worst-case computation time $W(a)$, or the robustness $R(a)$ relative to uncertainty of the inputs. Then, the actual optimality criterion that we use to select the optimal algorithm can no longer be reduced to a single numerical objective function, this criterion is more complex. Namely, in the resulting criterion, a is better than or of the same quality as a' (we will denote it by $a \geq a'$) if:

- either $A(a) < A(a')$,
- or $A(a) = A(a')$ and $W(a) \leq W(a')$.

If there are several algorithms which are optimal with respect to this new optimality criterion, then we can use the remaining non-uniqueness to optimize something else, and thus, get an even more complex optimality criterion.

This can continue until we finally get a criterion for which there is exactly one optimal alternative.

From this viewpoint, to define an optimality criterion, we should not restrict ourselves to numerical objective functions, we should have the most general definition.

No matter how complex the criterion, what we need is to be able to compare two different alternatives:

- either a is better than b ($a > b$),
- or b is better than a ($b > a$),
- or these two alternatives are of the same quality ($a \equiv b$).

Of course, this selection must be consistent: if a is better than b and b is better than c, then we should be able to conclude that a is better than c. In other words, the preference relation should be transitive.

From this viewpoint, it is reasonable to define an optimality criterion as a *pre-ordering relation*, i.e., a relation $a \geq b$ which is transitive and *reflexive* (i.e., $a \geq a$ for all a).

Which optimality criteria are reasonable?

First reasonable property of an optimality criterion for comparing different deep learning-related combination operations: the optimality criterion should be final. As we have mentioned earlier, if the optimality criterion selects several different alternatives as equally good, this means that this optimality criterion is not final: we still need to come up with an additional criterion for selecting one of these "optimal" alternatives. Selecting this additional criterion means that we modify the original optimality criterion $\geq$.

At the end, we should end up with a final criterion, for which there is only one optimal alternative.

Comment. It goes without saying that there should be at least one optimal alternative – otherwise, if no alternative is optimal, what should we choose?

Second reasonable property of an optimality criterion for comparing different deep learning-related combination operations: the optimality criterion should be scale-invariant. As we have mentioned earlier, the values v_i are usually obtained from re-scaling. Usually, we re-scale to the interval $[0, 1]$ by dividing all the values by the largest possible value of the corresponding quantity.

The resulting re-scaling is not unique: e.g., if we add one more quantity which is somewhat larger than what we have seen so far, then the maximum increases, and we need to re-scale the original values some more, i.e., replace the original values v_i with res-scaled values $\lambda \cdot v_i$.

In some cases, the values v_i are not values of the physical quantity but probabilities. In this case, the value are already in the interval $[0, 1]$. However, re-scaling is possible in this case as well. Namely, most probabilities that we deal with are *conditional* probabilities, and if we slightly change the condition, this leads to a re-scaling of the corresponding

probabilities. Indeed, in general, $P(A\,|\,B) = \dfrac{P(A\,\&\,B)}{P(B)}$. So, if $B \subset B'$, then for each event $A \subseteq B$, we have $P(A\,|\,B) = \dfrac{P(A)}{P(B)}$ and $P(A\,|\,B') = \dfrac{P(A)}{P(B')}$. Thus, if we replace the original condition B with the new condition B', then all conditional probabilities are re-scaled: $P(A\,|\,B') = \lambda \cdot P(A\,|\,B)$, where $\lambda \stackrel{\text{def}}{=} \dfrac{P(B)}{P(B')}$.

If instead of the original values a and b, we consider re-scaled values $a' = \lambda \cdot a$ and $b' = \lambda \cdot b$, then, instead of the combined value $a * b$, we get a new combined value $(\lambda \cdot a) * (\lambda \cdot b)$. We can re-scale it back into the old units, and get a new operation

$$a *_\lambda b = \lambda^{-1} \cdot ((\lambda \cdot a) * (\lambda \cdot b)).$$

This re-scaling should not affect the relative quality of different combination operations:

- if a combination operation $*$ was better than a combination operation $*'$, i.e., if we had $* > *'$,
- then after re-scaling, we should get the same preference: $*_\lambda > *'_\lambda$.

In this sense, the optimality criterion for comparing different deep learning-related combination operations should be *scale-invariant*.

This, we arrive at the following definitions.

8.3 Definitions and the Main Result

Definition 8.1. *By a* combination operation*, we mean a commutative, associative, continuous operation $a * b$ that transforms two non-negative real numbers a and b into a non-negative real number $a * b$ and which is (non-strictly) monotonic in each of the variables, i.e.:*

- *if $a \le a'$, then $a * b \le a' * b$, and*
- *if $b \le b'$, then $a * b \le a * b'$.*

Definition 3.2. *By a* reasonable optimality criterion*, we mean a pre-ordering (i.e., transitive and reflexive) relation $\geq$ on the set of all combination operations which is:*

- final*, in the sense that for this criterion, there exist only one optimal combination operation $*_{\rm opt}$ for which $\forall * \ (*_{\rm opt} \geq *)$; and*
- scale-invariant*: for every $\lambda > 0$, if $* \geq *'$, then $*_\lambda \geq *'_\lambda$, where*

$$a *_\lambda b \stackrel{\rm def}{=} \lambda^{-1} \cdot ((\lambda \cdot a) * (\lambda * b)).$$

Proposition 3.1. *For every reasonable optimality criterion, the optimal combination operation has one of the following forms: $a * b = 0$, $a * b = \min(a, b)$, $a * b = \max(a, b)$, and $a * b = (a^\alpha + b^\alpha)^{1/\alpha}$ for some α.*

Discussion. What are the "averaging" operations corresponding to these optimal combination operations?

For $a * b = 0$, the property $v_1 * \ldots * v_m = v * \ldots * v$ is satisfied for any possible v, so this combination operation does not lead to any "averaging" at all.

For $a * b = \min(a, b)$, the condition $v_1 * \ldots * v_m = v * \ldots * v$ leads to

$$v = \min(v_1, \ldots, v_m).$$

For $a * b = \max(a, b)$, the condition $v_1 * \ldots * v_m = v * \ldots * v$ leads to

$$v = \max(v_1, \ldots, v_m).$$

This "averaging" operation is actually sometimes used in deep learning – although not in dropout training [27].

Finally, for the combination operation $a * b = (a^\alpha + b^\alpha)^{1/\alpha}$, the condition $v_1 * \ldots * v_m = v * \ldots * v$ leads to

$$v = \left(\frac{v_1^\alpha + \ldots + v_m^\alpha}{m} \right)^{1/\alpha}.$$

For $\alpha = 1$, we get arithmetic average, and for $\alpha \to 0$, we get the geometric mean – the combination operation which turned out to be empirically the best for deep learning-related dropout training.

Indeed, in this case, the condition $v_1 * \ldots * v_m = v * \ldots * v$ takes the form

$$(v_1^\alpha + \ldots + v_m^\alpha)^{1/\alpha} = (v^\alpha + \ldots + v^\alpha)^{1/\alpha},$$

which is equivalent to

$$v_1^\alpha + \ldots + v_m^\alpha = m \cdot v^\alpha.$$

For every real value a, we have

$$a^\alpha = (\exp(\ln(a))^\alpha = \exp(\alpha \cdot \ln(a)).$$

For small x, $\exp(x) \approx 1 + x$, so $a^\alpha \approx 1 + \alpha \cdot \ln(a)$. Thus, the above condition leads to

$$(1 + \alpha \cdot \ln(v_1)) + \ldots + (1 + \alpha \cdot \ln(v_m)) = m \cdot (1 + \alpha \cdot \ln(v)),$$

i.e., to

$$m + \alpha \cdot (\ln(v_1) + \ldots + \ln(v_m)) = m + m \cdot \alpha \cdot \ln(v),$$

and thus, to

$$\ln(v) = \frac{\ln(v_1) + \ldots + \ln(v_m)}{m} = \frac{\ln(v_1 \cdot \ldots \cdot v_m)}{m};$$

hence to $v = \sqrt[m]{v_1 \cdot \ldots \cdot v_m}$.

So, we indeed have a 1-D family that contains combination operations efficiently used in deep learning:

- the arithmetic average that naturally comes from the use of the Least Squares optimality criterion, and
- the geometric mean, empirically the best combination operation for deep learning-related dropout training.

8.4 Proof

1°. Let us first prove that the optimal combination operation $*_{\rm opt}$ is scale-invariant, i.e., $(*_{\rm opt})_\lambda = *_{\rm opt}$ for all λ.

Indeed, let us take any λ and consider the combination operation $(*_{\rm opt})_\lambda$. By definition, $*_{\rm opt}$ is the optimal combination operation, so $*_{\rm opt} \geq *$ for all combination operations $*$. In particular, for every combination operation $*$, we have $*_{\rm op} \geq *_{\lambda^{-1}}$. Thus, by scale-invariance, we have $(*_{\rm opt})_\lambda \geq (*_{\lambda^{-1}})_\lambda = *$. So, $(*_{\rm opt})_\lambda$ is better than or of the same quality than any other combination operation $*$. This means that the combination operation $(*_{\rm opt})_\lambda$ is optimal.

However, our optimality criterion is reasonable hence final; thus, it has only one optimal combination operation. Hence, $(*_{\rm opt})_\lambda = *$.

By definition of the re-scaling operation $*_\lambda$, this means that

$$\lambda^{-1} \cdot ((\lambda \cdot a) * (\lambda \cdot b)) = a * b,$$

i.e., equivalently, that

$$(\lambda \cdot a) * (\lambda \cdot b) = \lambda \cdot (a * b). \tag{8.1}$$

2°. To complete the proof of the Proposition, we can now use our result – proven in [4] – that every combination operation * that satisfied the condition (3.1) has one of the following forms: $a * b = 0$, $a * b = \min(a, b)$, $a * b = \max(a, b)$, and $a * b = (a^\alpha + b^\alpha)^{1/\alpha}$ for some α.

The Proposition is thus proven.

8.5 Conclusions to Chapter 8

In many application areas, it is important to make accurate predictions of future events. At present, among all machine learning techniques, deep learning algorithms leads to the

most accurate predictions. However, this accuracy comes at a price – deep learning algorithms require much more computation time for training than any other machine learning techniques. To speed up the training, researchers have proposed the "dropout" idea:

- we train different patterns on different sub-networks on the neural network, and then
- we "average" the results.

Which averaging operation should we use? In many similar situations, a simple arithmetic average works the best – this can be explained by the fact that in many practical cases, the errors are normally distributed, and for normal distributions, arithmetic average is indeed provably the best averaging operation. So, researchers originally expected that arithmetic average should work the best in dropout training as well. Just in case, they also tried other statistics-motivated averaging operations. Surprisingly, it turned out that for deep learning-related dropout training, neither the arithmetic average not other statistics-motivated arithmetic operations work well. What works the best is the geometric mean, a mathematical operation that does not seem to have a direct statistical motivation.

In this chapter, we provide a theoretical explanation for this surprising empirical success of geometric means. Specifically:

- We first analyze what would be reasonable properties for a combination operation used in deep learning-related dropout training and what kind of optimality criteria are appropriate for selecting the best combination operation.
- After that, we prove that for all reasonable optimality criteria, the optimal combination operation belongs to a special 1-parametric family, a family that includes both the usual arithmetic mean and the empirically efficient geometric mean.

Chapter 9

What Is the Optimal Bin Size of a Histogram: An Informal Description

A natural way to estimate the probability density function of an unknown distribution from the sample of data points is to use histograms. The accuracy of the estimate depends on the size of the histogram's bins. There exist heuristic rules for selecting the bin size. In this chapter, we show that these rules indeed provide the optimal value of the bin size.

Comment. The contents of this chapter was published in [22].

9.1 Formulation of the Problem

Need to estimate pdfs. One of the most frequent ways to describe a probability distribution is by specifying its probability density function (pdf)

$$\rho(x) \stackrel{\text{def}}{=} \frac{dp}{dx} = \lim_{h \to 0} \frac{\text{Prob}(X \in [x, x+h])}{h}.$$

In many practical situations, all we know about a probability distribution is a sample of data points corresponding to this distribution. How can we estimate the pdf based on this sample?

Enter histograms. A natural way to estimate the limit when h tends to 0 is to consider the value of the ratio corresponding to some small h:

$$\rho(x) \approx \frac{\text{Prob}(X \in [x, x+h])}{h}.$$

To use this expression, we need to approximate the corresponding probabilities $\text{Prob}(X \in [x, x+h])$. By definition, the probability of an event is the limit of this event's frequency when the number of data points increases. In particular,

$$\text{Prob}(X \in [x, x+h]) = \lim_{n\to\infty} \frac{n([x, x+h])}{n},$$

where n is the overall number of data points and $n([x, x+h])$ denotes the number of data points within the interval $[x, x+h]$. Thus, as an estimate for the corresponding probability, we can get the frequency $f([x, x+h])$ of this event, i.e., the ratio

$$f([x, x+h]) \approx \frac{n([x, x+h])}{n}.$$

This idea leads to the following estimate for $\rho(x)$;

$$\rho(x) \approx \frac{f([x, x+h])}{h}.$$

This estimate, known as a *histogram* approximation, was first introduced by Karl Pearson in [61]; for details, see, e.g., [18, 31, 38].

In a histogram, the range of possible values of the corresponding quantity x is divided into intervals $[x_i, x_{i+1}]$ $(1 \le i \le k)$ called *bins*. In most practical cases, all the bins have the same width $h_i = x_{i+1} - x_i$, i.e., $h_1 = \ldots = h_k = h$ for some $h > 0$.

For each bin i, we then estimate (and plot) the frequency $f_i \stackrel{\text{def}}{=} \frac{n([x_i, x_{i+1}])}{n}$ with which the data points fall into this bin.

For values x from the corresponding interval, the probability density

$$\rho(x) \stackrel{\text{def}}{=} \frac{dp}{dx} = \lim_{h\to 0} \frac{\text{Prob}(X \in [x, x+h])}{h}$$

is approximated as the ratio

$$\rho_i = \frac{f_i}{h_i}.$$

Need to select a bin size. To form a histogram, we need to select the bin size h.

How bin sizes are selected now. In situations when we have an additional information about the corresponding probability distribution, we can formulate the bin selection problem as a precise optimization problem, and get the solution; see, e.g., [17, 67]. In many such cases, the optimal bin size $h_{\rm opt}$ decreases with the number n of data points as

$$h_{\rm opt} = {\rm const} \cdot \frac{s}{n^{1/3}},$$

where s is the "width" of the distribution – this can be the range of the interval at which $\rho(x)$ is positive, or, for distributions like Gaussian for which the pdf is never equal to 0, the difference between two quantiles.

In many practical situations, however, we do not have any additional information about the probability distribution – only the sample itself. In such situations, several heuristic rules have been proposed, most of them using the same dependence $h \sim \frac{s}{n^{1/3}}$; see, e.g., [18, 31, 38]. These heuristic rules are justified by three things:

- first, as we have mentioned, under certain conditions, these rules do provide provably optimal bin sizes; so it makes sense to assume that they are optimal in more general situations as well;
- second, the experience of using these rules shows that they, in general, work better than several previously proposed heuristic rules,
- third, under these rules, the two components of the pdf estimation error are approximately equal to each other, and equality of two error components is often an indication of optimality.

What we do in this chapter. In this chapter, we provide a somewhat stronger justification for the existing heuristic methods of selecting the bin size. Specifically, we provide (informal) arguments that the current heuristic rules indeed provide the optimal bin size – namely, the bin size for which the pdf approximation error is the smallest possible.

9.2 Which Bin Sizes Are Optimal: Analysis of the Problem and the Resulting Recommendation

Two reasons why a histogram is different from the pdf. To find the optimal bin size h_{opt}, we need to describe how the approximation error – i.e., the error with which the histogram approximates the actual pdf – depends on the bin size h. There are two reasons why the histogram is different from the pdf – and these reasons lead to two components of the approximation error:

- first, for each bin, for all the values x from this bin, the histogram provides the same value while the probability density function $\rho(x)$ has, in general, different values at different points x inside this bin;
- second, each estimate ρ_i is based on a finite sample, and it is well known that in statistics, estimates based on a finite sample are approximate.

Let us therefore estimate both components of the approximation error.

First component of the approximation error: approximation error caused by the finiteness of the bin size. The first error component is caused by the fact that for all points x from the i-th bin, we use the same approximating value ρ_i, while the actual pdf $\rho(x)$ is, in general, different for different points within the bin.

We can describe the corresponding approximation error as follows: instead of using the values $\rho(x)$ corresponding to different points x from the bin, we select one point x_0 from the bin, and use the value $\rho(x_0)$ instead of the actual value $\rho(x)$.

The larger the distance $|x - x_0|$ between the points x and x_0, the more the actual value $\rho(x)$ is different from our approximation $\rho(x_0)$. The worst case is when the difference $|x - x_0|$ is the largest possible. Thus, as x_0, we should select the point for which this largest distance is as small as possible.

One can easily check that this means selecting the midpoint $x_0 = x_{\text{mid}}$ of the bin. Indeed, in this case, the worst-case distance is equal to $s/2$, while if we select the point x_0

tilted to the left or to the right, the worst-case distance will be larger.

So, the first component of the approximation error comes from the difference $|\rho(x) - \rho(x_{\rm mid})|$ between the values of the pdf at the points x and $x_{\rm mid}$ for which $|x - x_{\rm mid}| = h/2$.

How big is this difference? Most practical distributions are unimodal: the corresponding pdf starts from 0, increases until it reaches its maximum value, and then decreases back to 0.

On the interval of the distribution width s, the pdf $\rho(x)$ goes from 0 to its maximum value $\rho_{\max}$ and back. We do not know which part of the interval of size s corresponds to increasing and which to decreasing. Since there is no reason to believe that the increasing part is longer than the decreasing one or vice versa, it make sense to assume, in our estimates, that these two parts has the same width, i.e., that the pdf increases on the interval of width $s/2$ and then decrease on the interval of the same width.

On the interval of size $s/2$, the value of the pdf $\rho(x)$ increases from 0 to its maximum value $\rho_{\max}$. The change of $\rho(x)$ on the interval of width $h/2$ should be proportional to this wirth, i.e., we should have

$$\Delta\rho \stackrel{\rm def}{=} |\rho(x) - \rho(x_{\rm mid})| \approx \frac{h/2}{s/2} \cdot \rho_{\max}.$$

Thus, the relative value $\dfrac{\Delta\rho}{\rho}$ of the first component of the approximation error is approximately equal to

$$\frac{h/2}{s/2} = \frac{h}{s}. \tag{9.1}$$

Second component of the approximation error: approximation error caused by the finiteness of the sample. The second component of the approximation error is cased by the fact that each estimate ρ_i is based on the finite sample. It is known (see, e.g., [18, 31, 38]) that when we estimate a parameter based on a sample of size m, we get an estimate with a relative error $\dfrac{1}{\sqrt{m}}$.

On the range of width s we have several bins of size h. Thus, the overall number of bins is equal to $k = s/h$. The overall number of data points is n, so in each of the k bins,

we have, on average,

$$m = \frac{n}{k} = \frac{n}{s/h} = \frac{n \cdot h}{s}$$

points. Based on these number of points, we get the following formula for the relative value of the second component of the approximation error:

$$\frac{1}{\sqrt{m}} = \frac{\sqrt{s}}{\sqrt{n \cdot h}}. \tag{9.2}$$

Overall approximation error. By adding the two components (9.1) and (9.2) of the approximation error, we get the following expression for the overall relative approximation error E:

$$E = \frac{h}{s} + \frac{\sqrt{s}}{\sqrt{n \cdot h}}. \tag{9.3}$$

Let us find the optimal bin size. To find the optimal bin size, we differentiate the expression (9.3) with respect to h and equate the resulting derivative to 0. As a result, we get

$$\frac{1}{s} - \frac{1}{2} \cdot \sqrt{\frac{s}{n}} \cdot h^{-3/2} = 0,$$

hence

$$\frac{1}{2} \cdot \sqrt{\frac{s}{n}} \cdot h^{-3/2} = \frac{1}{s}.$$

If we multiply both sides of this equality by $h^{3/2} \cdot s$, we conclude that

$$h^{3/2} = 2 \cdot \frac{s^{3/2}}{n^{1/2}}.$$

By raising both sides by the power 2/3, we get the formula

$$h_{\text{opt}} = \text{const} \cdot \frac{s}{n^{1/3}}.$$

This is exactly the heuristic rule that we wanted to justify.

Chapter 10

Conclusions

To speed up the quality assessment of the newly built roads, it is desirable to measure the road's stiffness in real time, as the road is being built. This is the main idea behind *intelligent compaction*, when accelerometers and other measuring instruments are attached to the roller and to other road-building equipment, and the results of the corresponding measurements are used to gauge the road's stiffness. The main challenge in implementing this idea is that the relation between the measured quantities (such as acceleration) and the desired quantities (such as elastic modulus that describes the road's stiffness) is very complicated, it is described by a complex system of partial differential equations which are difficult to solve in real time. It is therefore desirable to come up with easier-to-compute algorithms for estimating the road's quality based on the real-time measurement results.

The main task of this dissertation was the design of such algorithms. As a solution to this task, we propose both most-easy-to-compute analytical expressions and somewhat more complex (but still easy to compute) neural network models.

We also provide a theoretical explanation for the empirical formulas used to describe the road dynamics, and a theoretical explanation for the empirical safety factor that related the simulation results with actual road measurements.

In the process of solving the main task, we have also solved several auxiliary tasks whose solutions are of potential interest to more general data processing situations. The solutions to these auxiliary tasks explain:

- how many simulations are needed,
- what is the best relation between training and testing sets,

- how to take into account that we often need to minimize relative error,
- how to best apply neural networks, and
- what is the optimal bin size in a histogram.

We hope that both our solution of the main tasks and our solutions to the auxiliary tasks will be useful to practitioners.

References

[1] American Association of State Highway and Transportation Officials (AASHTO), "Resilient Modulus of Subgrade Soils and Untreated Base/Subbase Materials", Standard T 292-91, In: *Standard Specifications for Transportation Materials and Methods of Sampling and Testing*, AAHSTO, Washington, DC, 1998, pp. 1057–1071.

[2] Alpaydin, E. *Introduction to Machine Learning*, The MIT Press, Cambridge, Massachusetts, 2004.

[3] Anderegg, R. and Kaufmann, K. "Intelligent Compaction with Vibratory Rollers: Feedback Control Systems in Automatic Compaction and Compaction Control", *Transportation Research Record*, 2004, Vol. 1868, pp. 124–134.

[4] Autchariyapanitkul, K., Kosheleva, O., Kreinovich, V., and Sriboonchitta, S. "Quantum econometrics: how to explain its quantitative successes and how the resulting formulas are related to scale invariance, entropy, and fuzziness", In: Huynh, V.-N., Inuiguchi, M., Tran, D.-H., and Denoeux, T. (eds.), *Proceedings of the International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making IUKM'2018*, Hanoi, Vietnam, March 13–15, 2018.

[5] Baral, C., Fuentes, O., and Kreinovich, V. "Why deep neural networks: a possible theoretical explanation", In: Ceberio, M., and Kreinovich, V. (eds.), *Constraint Programming and Decision Making: Theory and Applications*, Springer Verlag, Berlin, Heidelberg, 2018, pp. 1–6.

[6] Barragan, P., Nazarian, S., Kreinovich, V., Gholamy, A., and Mazari, M. "How to Estimate Resilient Modulus for Unbound Aggregate Materials: A Theoretical Explanation of an Empirical Formula", *Proceedings of the 2016 World Conference on Soft Computing*, Berkeley, California, May 22–25, 2016, pp. 203–207.

[7] Beliakov, G., Pradera, A., and Calvo, T. *Aggregation Functions: A Guide for Practitioners*, Springer Verlag, Berlin, Heidelberg, 2007.

[8] Berger, J.R. "Boundary Element Analysis of Anisotropic Bimaterials with Special Green's Functions", *Journal of Engineering Analysis with Boundary Elements*, 1995, Vol. 14, pp. 123–131.

[9] Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, New York, 2006.

[10] Bouchon-Meunier, B. (ed.), *Aggregation and Fusion of Imperfect Information*, Physica Verlag, Berlin, Heidelberg, 2013.

[11] Brebbia, C.A. and Dominguez, J. *Boundary Elements: An Introductory Course*, McGraw-Hill, UK, 1992.

[12] Bustince, H., Fernandez, J., Mesiar, R., and Calvo, T. (eds.), *Aggregation Functions in Theory and in Practice*, Springer Verlag, Berlin, Heidelberg, 2013.

[13] Calvo, T., Mayor, G., and Mesiar, R. (eds.), *Aggregation Operators: New Trends and Applications*, Physica Verlag, Heidelberg, New York, 2002.

[14] Chang, G., Xu, Q. Rutledge, J. Horan, R. Michael, L. White, D. and Vennapusa, P. *Accelerated Implementation of Intelligent Compaction Technology for Embankment Subgrade Soils, Aggregate Base, and Asphalt Pavement Materials*, Report FHWA-IF-12-002. Federal Highway Administration, Washington, D.C., 2011.

[15] Facas, N.W., van Susante, P.J., and Mooney, M.A. "Influence of Rocking Motion on the Vibratory Roller-Based Measurement of Soil Stiffness," *ASCE Journal of Engineering Mechanics*, 2010, Vol. 136, No. 7, pp. 898–905.

[16] Feynman, R., Leighton, R., and Sands, M. *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.

[17] Freedman, D., and Diaconis, P. "On the histogram as a density estimator: L_2 theory", *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 1981, Vol. 57, No. 4, pp. 453–476.

[18] Freedman, D., Pisani, R., and Purves, R. *Statistics*, W. W. Norton, New York, 1998.

[19] Fuentes, O., Parra, J., Anthony, E., and Kreinovich, V. "Why rectified linear neurons are efficient: a possible theoretical explanations", In: Kosheleva, O., Shary, S., Xiang, G., and Zapatrin, R. (eds.), *Beyond Traditional Probabilistic Data Processing Techniques: Interval, Fuzzy, etc. Methods and Their Applications*, Springer, Cham, Switzerland, to appear.

[20] Gholamy, A., *Backcalculation of Intelligent Compaction Data for the Mechanical Properties of Soil Geosystems*, PhD Dissertation Proposal, University of Texas at El Paso, 2018.

[21] Gholamy, A., and Kreinovich, V., "How to Use Absolute-Error-Minimizing Software to Minimize Relative Error: Practitioner's Guide", *International Mathematical Forum*, 2017, Vol. 12, No. 16, pp. 763–770.

[22] Gholamy, A., and Kreinovich, V., "What Is the Optimal Bin Size of a Histogram: An Informal Description", *International Mathematical Forum*, 2017, Vol. 12, No. 15, pp. 731–736.

[23] Gholamy, A., and Kreinovich, V., "How Many Monte-Carlo Simulations Are Needed to Adequately Process Interval Uncertainty: An Explanation of the Smart Electric Grid-Related Simulation Results", *Journal of Innovative Technology and Education*, 2018, Vol. 5, No. 1, pp. 1–5.

[24] Gholamy, A., and Kreinovich, V., "Safety Factors in Soil and Pavement Engineering: Theoretical Explanation of Empirical Data", *Abstracts of the 23rd Joint UTEP/NMSU*

Workshop on Mathematics, Computer Science, and Computational Sciences, El Paso, Texas, November 3, 2018.

[25] Gholamy, A., Kreinovich, V., and Kosheleva, O. "Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation", *International Journal of Intelligent Technologies and Applied Statistics*, 2018, Vol. 11, No. 2, pp. 105–111.

[26] Gholamy, A., Parra, J., Kreinovich V., Fuentes, O., and Anthony, E. "How to Best Apply Deep Neural Networks in Geosciences: Towards Optimal 'Averaging' in Dropout Training", In: Watada, J., Tan, S.C., Vasant, P., Padmanabhan, E., and Jain, L.C. (eds.), *Smart Unconventional Modelling, Simulation and Optimization for Geosciences and Petroleum Engineering*, Springer Verlag, to appear.

[27] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Leaning*, MIT Press, Cambridge, Massachusetts, 2016.

[28] Grabisch, M., Marichal, J.-L., Mesiar, R., and Pap, E. *Aggregation Functions*, Cambridge University Press, Cambridge, UK, 2009.

[29] Göktepe, A.B., Agar, E., and Lav, A.H. "Comparison of Multilayer Perceptron and Adaptive Neuro-Fuzzy System on Backcalculating the Mechanical Properties of Flexible Pavements." *The Bulletin of the Istanbul Technical University ARI*, 2005, Vol. 54, No. 3, pp. 65–67.

[30] Göktepe, A.B., Agar, E., and Lav, A. H. "Advances in Backcalculating the Mechanical Properties of Flexible Pavements." *Advances in Engineering Software*, 2006, Vol. 37, No. 7, pp. 421–431.

[31] Howitt, D., and Cramer, D. *Statistics in Psychology*, Prentice Hall, Upper Saddle River, New Jersey, 2008.

[32] Jaulin, L., Kiefer, M., Didrit, O., and Walter, E. *Applied Interval Analysis, with*

Examples in Parameter and State Estimation, Robust Control, and Robotics, Springer, London, 2001.

[33] Huber, P.J., and Ronchetti, E.M. *Robust Statistics*, Wiley, Hoboken, New Jersey, 2009.

[34] Kang Y.V. "Multi-Frequency Backcalculation of Pavement Layer Moduli", *Journal of Transportation Engineering,* 1998, Vol. 124, No. 1, pp. 73–81.

[35] Kargah-Ostadi, N., and Stoffels, S.M. "Backcalculation of flexible pavement structural properties using a restart covariance matrix adaptation evolution strategy", *Journal of Computing in Civil Engineering*, 2013, Paper 04014035.

[36] Kreinovich, K., Lakeyev, A., Rohn, J., and Kahl, P. *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1998.

[37] Kröber, W., Floss, R., and Wallrath, W. *Dynamic Soil Stiffness as Quality Criterion for Soil Compaction. Geotechnics for Roads, Rail Tracks and Earth Structures*, A.A. Balkema Publishers, Leiden, The Netherlands, 2001.

[38] Lancaster, H.O. *An Introduction to Medical Statistics*, John Wiley and Sons, New York, 1974.

[39] Levasseur, S., Malécot, Y., Boulon, M., and Flavigny, E. "Soil Parameter Identification using a Genetic Algorithm", *International Journal for Numerical and Analytical Methods in Geomechanics*, 2008, Vol. 32, No. 2, pp. 189–213.

[40] Loia, V., Terzija, V., Vaccaro, A., and Wall, P. "An affine-arithmetic-based consensus protocol for smart-grid computing in the presence of data uncertainties", *IEEE Transactions on Industrial Electronics*, 2015, Vol. 62, No. 5, pp. 2973–2982.

[41] Lorkowski, J., and Kreinovich, V. "How to Gauge Unknown Unknowns: A Possible Theoretical Explanation of the Usual Safety Factor of 2", *Mathematical Structures and Modeling*, 2014, Vol. 32, pp. 49–52.

[42] Maina, J.W., Yokota, H., Mfinanga, D.A., and Masuda, S. "Prediction of Pavement Deterioration based on FWD Results", In: Tayabji, S.D., and Lukanen, E.O. (eds.), *NDT of Pavements and Backcalculation of Moduli, Vol. 3*, ASTM Special Technical Report STP 1375, 2000, pp. 95–109.

[43] Mayer, G. *Interval Analysis and Automatic Result Verification*, de Gruyter, Berlin, 2017.

[44] Mazari, M., Navarro, E., Abdallah, I., and Nazarian, S. "Comparison of numerical and experimental responses of pavement systems using various resilient modulus models", *Soils and Foundations*, 2014, Vol. 54, No. 1, pp. 36–44.

[45] Mazari M., Tirado, C., Abdallah, I., and Nazarian, S. "Mechanical estimation of lightweight deflectometer target field modulus for construction quality control", *Geotechnical Testing Journal*, 2016, doi:10.1520/GTJ20150266.

[46] Meier, R. and Alexander, D. "Using ANN as a Forward Approach to Back-calculation", *Transportation Research Record*, 1997, Vol. 1570, pp. 126–133.

[47] Meier, R.W. and Rix, G.J. "Backcalculation of Flexible Pavement Moduli using Artificial Neural Networks", *Transportation Research Record: Journal of the Transportation Research Board*, 1994, Vol. 1448, pp. 75–82.

[48] Meshkani, A., Abdallah, I., and Nazarian, S. *Determination of Nonlinear Parameters of Flexible Pavement Layers from Nondestructive Testing*, Research Report 1780-3, Center for Highway Materials Research, The University of Texas at El Paso, 2002.

[49] Mooney, M.A. and Adam, D. "Vibratory Roller Integrated Measurement of Earthwork Compaction: An Overview", *Proceedings of the 7th International ASCE Symposium on Field Measurements in Geomechanics*, Boston, Massachusetts, September 24–27, 2007.

[50] Mooney, M.A., and Facas, N.W. *Extraction of Layer Properties from Intelligent Compaction Data*, Final Report for NCHRP Highway IDEA Project 145, Transportation Research Board of the National Academies, Washington, DC, 2013.

[51] Mooney, M.A. and Rinehart, R.V. "Field Monitoring of Roller Vibration during Compaction of Subgrade Soil", *Journal of Geotechnical and Geoenvironmental Engineering*, 2007, Vol. 133, No. 3, pp. 257–265.

[52] Mooney, M.A. and Rinehart, R.V. "Instrumentation of a Roller Compactor to Monitor Vibration Behavior during Earthwork Compaction", *Journal of Automation in Construction*, 2008, Vol. 17, No. 2, pp. 144–150.

[53] Mooney, M.A., Rinehart, R.V., White, D.J., Vennapusa, P., Facas, N., and Musimbi, O. *Intelligent Soil Compaction Systems*, NCHRP Report 676. Transportation Research Board, Washington, DC, 178 pp., 2011.

[54] Moore, R.E., Kearfott, R.B., and Cloud, M.J. *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[55] Musimbi, O.M. *Experimental and Numerical Investigation of Vibratory Drum Interacting with Layered Elastic Media*, PhD Dissertation, Division of Engineering, Colorado School of Mines, 2011.

[56] Nguyen, H.T., and Kreinovich, V. *Applications of continuous mathematics to computer science*, Kluwer, Dordrecht, 1997.

[57] Ooi, P.S.K., Archilla, R., and Sandefur, K.G. "Resilient modulus models for compacted cohesive soils", *Transportation Research Record*, 2006, No. 1874, pp. 115–124.

[58] Parra, J., Fuentes, O., Anthony, E., and Kreinovich, V. "Prediction of volcanic eruptions: case study of rare events in chaotic systems with delay", *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics SMC'2017,* Banff, Canada, October 5–8, 2017.

[59] Parra, J., Fuentes, O., Anthony, E., and Kreinovich, V. "Use of machine learning to analyze and – hopefully – predict volcano activity", *Acta Politechnica Hungarica*, 2017, Vol. 14, No. 3, pp. 209–221.

[60] Parra, J., Fuentes, O., Kreinovich, V., Anthony, E., Espejel, V., and Hinojosa, O. "Eruption forecasting from seismic activity using neural networks", *Proceedings of International Association of Vulcanology and Chemistry of the Earth's Interior (IAVSEI) Scientific Assembly IAVCEI'2017 "Fostering Integrative Studies of Volcanism"*, Portland, Oregon, August 14–17, 2017.

[61] Pearson, K. "Contributions to the mathematical theory of evolution.II. Skew variation in homogeneous material", *Philosophical Transactions of the Royal Society A: Mathematical, Physical, and Engineering Sciences*, 1895, Vol. 186, pp. 343–414.

[62] Rabinovich, S.G. *Measurement Errors and Uncertainty: Theory and Practice*, Springer Verlag, Berlin, 2005.

[63] Rinehart, R.V., Berger, J.R., and Mooney, M.A. "Comparison of stress states and paths: vibratory roller-measured soil stiffness and resilient modulus testing", *Transportation Research Record*, 2009, Vol. 2116, pp. 8–15.

[64] Rinehart, R.V., Mooney, M.A., and Berger, J.R. "In-Ground Stress-Strain beneath Center and Edge of Vibratory Roller Compactor", *Proceedings of the 1st International Conference on Transportation Geotechnics*, Nottingham, U.K., August 25–27, 2008.

[65] Roesset, J.M., and Shao, K.Y. "Dynamic Interpretation of Dynaflect and Falling Weight Deflectometer Tests", *Transportation Research Record: Journal of the Transportation Research Board*, 1985, Vol. 1022, pp. 7–16.

[66] Sadd, M.H. *Elasticity: Theory, Applications, and Numerics*, Academic Press, Oxford, UK, and Waltham, Massachusetts, 2014.

[67] Scott, D.W. "Averages shifted histogram", *Wiley Interdisciplinary Reviews: Computational Statistics*, 2009, Vol. 2, No. 2, pp. 160–164.

[68] Sheskin, D.S. *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.

[69] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhudinov, R. "Dropout: a simple way to prevent neural networks from overfitting", *Journal of Machine Learning Research*, 2014, Vol. 15, pp. 1929–1958.

[70] Szegedy, C., Liu, W., Jia, Y., Serament, P., Reed, S., Anguelov, D., Erhan, E., Vanhoucke, V., and Rabinovich, A. *Going Deeper with Convolution*. arXiv:1409.4842 (2014)

[71] Torra, V., Mesiar, R., and De Baets, B. (eds.), *Aggregation Functions in Theory and in Practice*, Springer, Cham, Switzerland, 2018.

[72] Ullidtz, P. "Will Nonlinear Backcalculation Help?", In: Tayabji, S.D., and Lukanen, E.O. (eds.), *NDT of Pavements and Backcalculation of Moduli, Vol. 3*, ASTM Special Technical Publication STP 1375, 2000, pp. 14–22.

[73] Uzan, J. "Advanced Backcalculation Techniques." In: Von Quintus, H.L., Bush, A.J., and Baladi, G.Y. (eds.), *NDT of Pavements and Backcalculation of Moduli*, ASTM Special Technical Publication STP 1198, 1994, pp. 3–37.

[74] Van Susante, P.J. and Mooney, M.A. "Capturing Nonlinear Vibratory Roller Compactor Behavior through Lumped Parameter Modeling", *ASCE Journal of Engineering Mechanics*, 2008, Vol. 134, No. 8, pp. 684–693.

[75] Warde-Farley, D., Goodfellow, J.J., Courville, A., and Bendigo, Y. "An empirical analysis of dropout in piecewise linear networks", *Proceedings of the 2nd International Conference on Learning Representations ICLR'2014*, Banff, Canada, April 14–16, 2014.

[76] Yoo, T.-S. and Selig, E.T. "Dynamics of Vibratory-Roller Compaction", *ASCE Journal of the Geotechnical Engineering Division*, 1979, Vol. 105, No. GT10, pp. 1211–1231.

Curriculum Vitae

Afshin Gholamy earned his Bachelor of Engineering degree in Mining Engineering from University of Tehran in 2005. In 2010, he received his Master of Science degree in Exploration Seismology from the University of Tehran. He earned his second Master of Science degree in Geophysics from the University of Texas at El Paso (UTEP) in 2014 and joined UTEP's doctoral program in Geophysics in 2015.

While being a doctoral candidate at UTEP, Afshin Gholamy received several awards and scholarship/fellowships, including UTEP Graduate School Frank B. Cotton Trust Scholarship and Vernon G. and Joy Hunt Endowed Scholarship. He was also a recipient of the Permian Basin Geophysical Society (PBGS) Award for three consecutive years between 2016 and 2018. He also earned the Geological Department Graduate Student Excellence Endowed Scholarship in 2018.

Dr. Gholamy's research results were presented at several meetings and conferences including the 2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES), 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA/EUSFLAT), 2015 Society of Exploration Geophysicists (SEG) Annual Meeting, 2016 World Conference on Soft Computing, 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), and 2018 Transportation Research Board (TRB) Annual Meeting. His work has appeared in the peer-refereed proceedings of the aforementioned conferences as well as in several academic journals. He has also several articles forthcoming soon.

While pursuing his degree, Dr. Gholamy worked as a Teaching/Research Assistant for the Department of Geological Sciences as well as Research Associate for the Department of Civil Engineering. He also served as a field geophysicist in several nationally and internationally recognized research experiments such as IMUSH and ENAM.

Permanent address: 4111 Westcity Court Apt. 166, El Paso, Texas 79902