Pontifícia Universidade Católica
do Rio de Janeiro

Luiz Alberto Barbosa de Lima

Porosity Estimation from Seismic Attributes with Simultaneous Classification of Spatially Structured Latent Facies

Tese de Doutorado

Thesis presented to the Programa de Pós-graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica.

Advisor: Profa. Marley Maria Bernardes Rebuzzi Vellasco

Rio de Janeiro

February 2017

Pontifícia Universidade Católica
do Rio de Janeiro

Luiz Alberto Barbosa de Lima

Porosity Estimation from Seismic Attributes with Simultaneous Classification of Spatially Structured Latent Facies

Thesis presented to the Programa de Pós-Graduação em Engenharia Elétrica of PUC-Rio, in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica. Approved by the undersigned Examination Committee.

Prof.ª Marley Maria Bernardes Rebuzzi Vellasco
Advisor
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Marcelo Gattass
Departamento de Informática – PUC-Rio

Prof. Ruy Luiz Milidiu
Departamento de Informática – PUC-Rio

Prof. Joao Felipe Coimbra Leite Costa
UFRGS

Luiz Eduardo Seabra Varella
Petrobras-Petroleo Brasileiro SA

Prof. Valmir Carneiro Barbosa
UFRJ

Prof. Márcio da Silveira Carvalho
Vice Dean of Graduate Studies
Centro Técnico Científico – PUC-Rio

Rio de Janeiro, February 17th, 2017

Luiz Alberto Barbosa de Lima

Graduated in Electric Engineering at the Universidade Federal do Rio de Janeiro (Rio de Janeiro, Brazil); MSc in Computer Science at the University of North Carolina at Chapel Hill (North Carolina, USA); Specialization in Business Intelligence at the Pontifícia Universidade Católica do Rio de Janeiro (Rio de Janeiro, Brazil).

Bibliographic data

Lima, Luiz Alberto Barbosa de

Porosity Estimation from Seismic Attributes with Simultaneous Classification of Spatially Structured Latent Facies / Luiz Alberto Barbosa de Lima; advisor: Marley Maria Bernardes Rebuzzi Vellasco. – Rio de Janeiro: PUC-Rio, Departamento de Engenharia Elétrica, 2017.

v., 98 f.: il. ; 29,7 cm

1. Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui bibliografia

1. Engenharia elétrica – Teses. 2. Porosity estimation. 3. Geological facies classification. 4. Conditional random field. 5. Latent variables. 6. Semi-supervised learning. 7. Transductive learning. I. Vellasco, Marley Maria Bernardes Rebuzzi. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 621.3

Acknowledgements

I would like to take this opportunity to gratefully acknowledge the assistance and contributions of many people throughout my PhD.

First, I would like to thank my advisor, Professor Marley Vellasco, whose guidance and support made this PhD possible. She was the first one to insist that I should pursue a PhD and since then she has always been there for me. I am really grateful for her knowledge sharing, enthusiasm and patience during all these years.

I would also like to thank Professor Klaus-Robert Müller for the great opportunity of spending two years as an external PhD student at his Intelligent Data Analysis lab (IDA) at the Technical University Berlin. His supervision, high energy and motivation were fundamental in my work. I extend my gratitude to all members of the IDA group and, in special, to Nico Görnitz and Shin Nakagima for their constant guidance, help with technical excellence and out-of-the-box thinking, and mainly, their most precious friendship.

My deep gratitude goes also to Professor Marcelo Gattass, for helping me in so many occasions and for being not only a mentor, but a truly friend.

I also would like to thank Professor Ruy Milidiú for opening my eyes to a new world of possibilities with structured learning.

My thanks extend to the Examination Committee members Marley Vellasco, Marcelo Gattass, Ruy Milidiú, João Felipe Costa, Luiz Eduardo Varella and Valmir Barbosa for all their insightful comments and remarks.

I would like to register a special thank to Fernando Rodrigues, Sebastião Pereira, Carlos Henriques Cunha, and Sylvia Anjos for providing me this great opportunity to pursue a PhD.

My appreciation also extends to Silmara Campos and Otaviano Pessoa Neto for all their help and support during the last stage of my PhD.

I am also extremely grateful to the TGEO team at Petrobras for all their support throughout this work. In particular, I would like to thank Luiz Eduardo Varella not only for helping me in many moments with his well known technical excellence, but also with his most valued friendship. I would like to thank as well Fabio Lima, Jacilene Torres, Lilian Montano, and Lucia Fonseca for their assistance on many occasions.

I thank Petrobras for their financial support.

Finally, I would like to thank my family, in special Clara, Isabela, Eduardo, Antônia, and Locke for all their love, patience, help, and support, mainly during the difficult times!

Abstract

Lima, Luiz Alberto Barbosa de; Vellasco, Marley Maria Bernardes Rebuzzi (Advisor). **Porosity Estimation from Seismic Attributes with Simultaneous Classification of Spatially Structured Latent Facies**. Rio de Janeiro, 2017. 98p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Estimating porosity in oil and gas reservoirs is a crucial and challenging task in the oil industry. A novel nonlinear model for porosity estimation is proposed, which handles sedimentary facies as latent variables. It successfully combines the concepts of conditional random fields (CRFs), transductive learning and ridge regression. The proposed *Transductive Conditional Random Field Regression (TCRFR)* uses seismic impedance volumes as input information, conditioned on the porosity values from the available wells in the reservoir, and simultaneously and automatically provides as output the porosity estimation and facies classification in the whole volume. The method is able to infer the latent facies states by combining the local, labeled and accurate porosity information available at well locations with the plentiful but imprecise impedance information available everywhere in the reservoir volume. That accurate information is propagated in the reservoir based on conditional random field probabilistic graphical models, greatly reducing uncertainty. In addition, two new techniques are introduced as preprocessing steps for the application of TCRFR in the extreme but realistic cases where just a scarce amount of porosity labeled samples are available in a few exploratory wells, a typical situation for geologists during the evaluation of a reservoir in the exploration phase. Both synthetic and real-world data experiments are presented to prove the usefulness of the proposed methodology, which show that it outperforms previous automatic estimation methods on synthetic data and provides a comparable result to the traditional manual labored geostatistics approach on real-world data.

Keywords

Porosity estimation; Geological facies classification; Conditional random field; Latent variables; Semi-supervised learning; Transductive learning.

Resumo

Lima, Luiz Alberto Barbosa de; Vellasco, Marley Maria Bernardes Rebuzzi (Orientadora). **Predição de Porosidade a partir de Atributos Sísmicos com Classificação Simultânea de Facies Geológicas Latentes em Estruturas Espaciais**. Rio de Janeiro, 2017. 98p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Predição de porosidade em reservatórios de óleo e gás representa em uma tarefa crucial e desafiadora na indústria de petróleo. Neste trabalho é proposto um novo modelo não-linear para predição de porosidade que trata fácies sedimentares como variáveis ocultas ou *latentes*. Esse modelo, denominado Transductive Conditional Random Field Regression (TCRFR), combina com sucesso os conceitos de *Markov random fields*, *ridge regression* e aprendizado transdutivo. O modelo utiliza volumes de impedância sísmica como informação de entrada condicionada aos valores de porosidade disponíveis nos poços existentes no reservatório e realiza de forma simultânea e automática a classificação das fácies e a estimativa de porosidade em todo o volume. O método é capaz de inferir as fácies latentes através da combinação de amostras precisas de porosidade local presentes nos poços com dados de impedância sísmica ruidosos, porém disponíveis em todo o volume do reservatório. A informação precisa de porosidade é propagada no volume através de modelos probabilísticos baseados em grafos, utilizando *conditional random fields*. Adicionalmente, duas novas técnicas são introduzidas como etapas de pré-processamento para aplicação do método TCRFR nos casos extremos em que somente um número bastante reduzido de amostras rotuladas de porosidade encontra-se disponível em um pequeno conjunto de poços exploratórios, uma situação típica para geólogos durante a fase exploratória de uma nova área. São realizados experimentos utilizando dados de um reservatório sintético e de um reservatório real. Os resultados comprovam que o método apresenta um desempenho consideravelmente superior a outros métodos automáticos de predição em relação aos dados sintéticos e, em relação aos dados reais, um desempenho comparável ao gerado por técnicas tradicionais de geoestatística que demandam grande esforço manual por parte de especialistas.

Palavras-chave

Estimativa de porosidade; classificação de facies geologicas; conditional random field; variáveis latentes; aprendizado semi-supervisionado; aprendizado transdutivo.

Summary

Figure list

Table list

List of Abbreviations

ARI – Adjusted Rand Index
BP – Belief Propagation
CCCP – Convex-Concave Procedure
CRF – Conditional Random Field
I.I.D. – Independent and identically distributed
L-BFGS - Limited-memory Broyden–Fletcher–Goldfarb–Shanno
LBP – Loopy Belief Propagation
LBPA – Loopy Belief Propagation Approximation
MAE – Mean Absolute Error
MAP – Maximum a Posteriori
MDAE – Median Absolute Error
MSE – Mean Squared Error
MRF – Markov Random Field
QPA – Quadratic Programming Approximation
RMSE – Root Mean Squared Error
R2 – Coefficient of determination
TCRFR – Transductive Conditional Random Field Regression

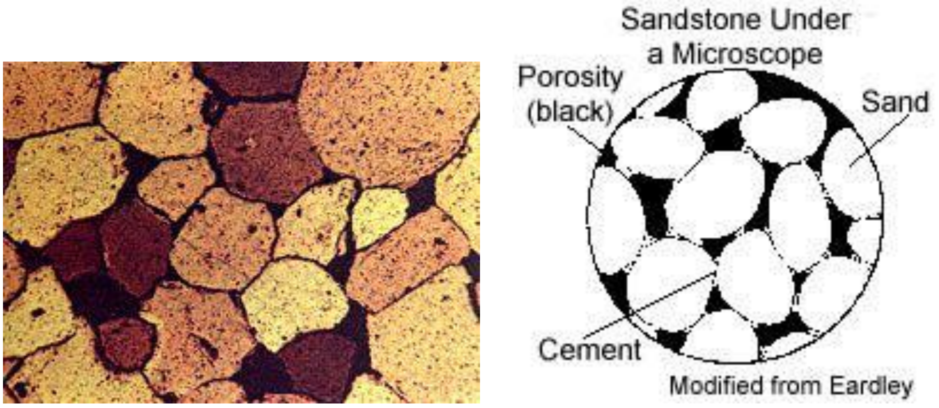
We are drowning in information but starved for knowledge.

John Naisbitt, *Megatrends, 1982.*

1 Introduction

1.1 Motivation

Porosity estimation in a reservoir is a critical task in the oil industry. Porosity is a key petrophysical property in the exploration and recovery of hydrocarbons like crude oil and natural gas, being fundamental in many different stages, like during the evaluation of rock formations by geologists, pore pressure assessment for well drilling, and also as an input parameter in flow simulations executed by reservoir engineers. It is defined as the fraction, or percentage, of void space over the total rock volume, where those hydrocarbons can be stored [8]. Figure 1.1(a) shows an actual sandstone slice, thinner than a human hair, seen from a microscope [1]. Sandstones belong to a class of reservoir rocks that usually have high porosity. *High* in this case means something commonly in the range of 8 to 15%. The larger brown and yellow grains are made of quartz and between the grains it is possible to observe the porosity void spaces. Porosity is also shown in black in Figure 1.1(b). These void spaces can be filled with oil and gas.

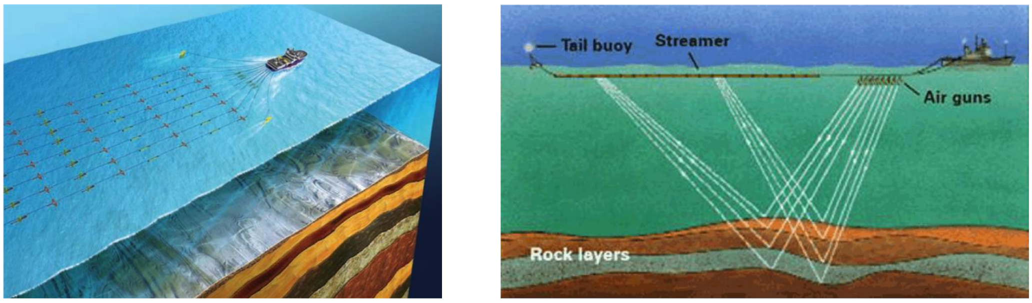


(a) Sandstone slice

(b) Sandstone schematics

Figure 1.1: Sanstone under microscope [1].

Porosity can be directly measured at wells once they are drilled, however drilling is extremely costly (tens of million dollars offshore) and typically conducted only at the locations where a petroleum reservoir is highly likely to exist. It is, therefore, often predicted from much less expensive, but also imprecise, indirect information sources. Seismic data constitutes one of the main sources of information used in reservoir analysis, being available in the whole reservoir volume. Seismic acquisition is an indirect method which relies upon data obtained from arrangements of sensors known as geophones (onshore) or hydrophones (offshore). A source, such as dynamite shot or an air gun, generates acoustic or elastic vibrations that travel into the Earth, pass through strata with different seismic responses and filtering effects, and return to the surface to be recorded as seismic data by those sensors [2]. Figure 1.2 shows an example of offshore seismic acquisition, extracted from [2]. Fig. 1.2(a) shows a seismic acquisition ship pulling an arrangement of hydrophone sensors. In Fig. 1.2(b) acoustic waves are emitted from an air gun, being reflected and refracted at different sedimentary rock layers.



(a) Seismic ship pulling hydrophones (b) Seismic acquisition

Figure 1.2: Offshore seismic acquisition process [2].

The seismic impedance volumes used as input information in all experiments in this thesis were obtained after the application of a succession of mathematical procedures on the original seismic reflection volume.

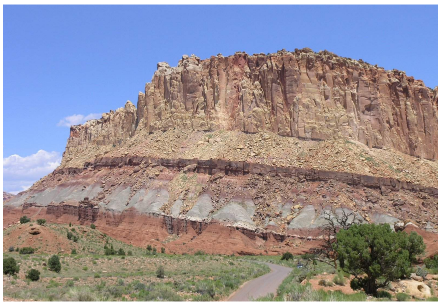
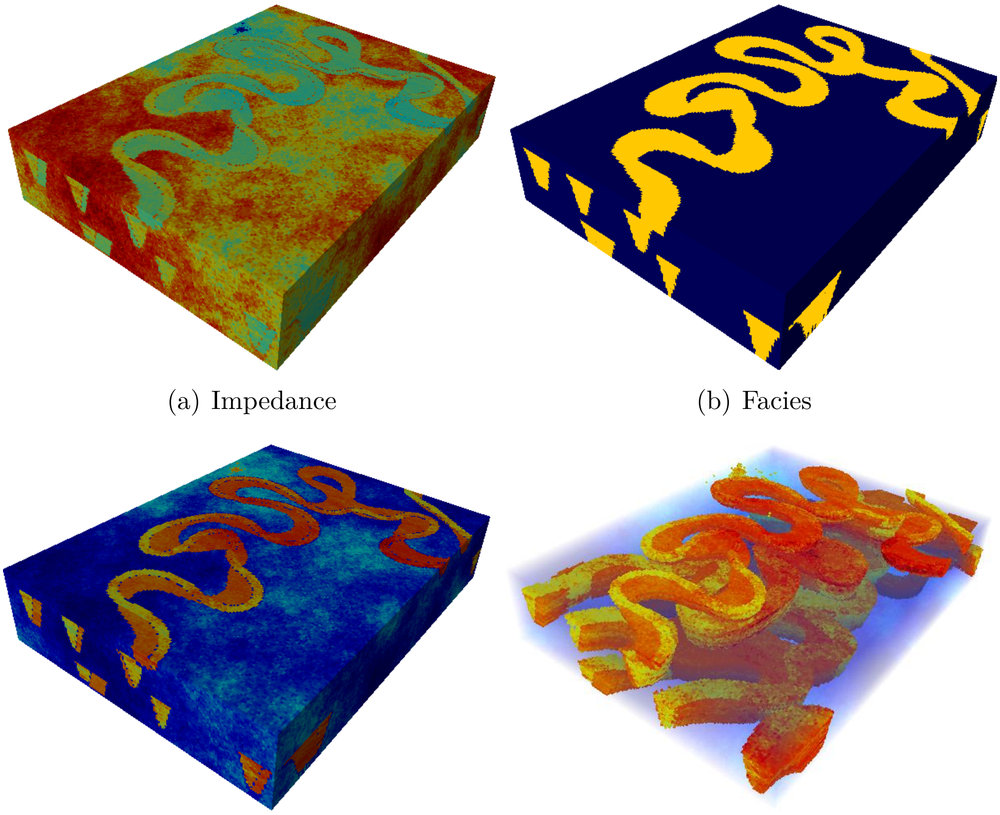


Figure 1.3: Example of distinct geological facies on a cliff [3].

There is an inverse correlation between seismic impedance and porosity, which heavily depends on latent sedimentary discontinuities known as *facies*. Each geological facies category represents a distinct group of rocks with similar features, like mineralogy and grain size. Figure 1.3 shows an example of clearly distinct geological facies, like horizontal packages of distinct rocks, on a cliff in the Capitol Reef National Park, CO, USA [3].

Figure 1.4 depicts different data views from the middle layer of the synthetic Stanford VI reservoir data set, adapted from [4]. Fig. 1.4(a) shows a synthetic impedance volume covering an area equivalent to approximately 20 square kilometers and 80 meters in depth. Fig. 1.4(b) shows the related facies. The corresponding porosity volume is illustrated in Fig. 1.4(c). The high porosity sandstone channels in the reservoir are highlighted in Fig. 1.4(d) and it is possible to observe the diverse channel geometrical shapes that occur in the reservoir, depending on the depth location.



(a) Impedance

(b) Facies

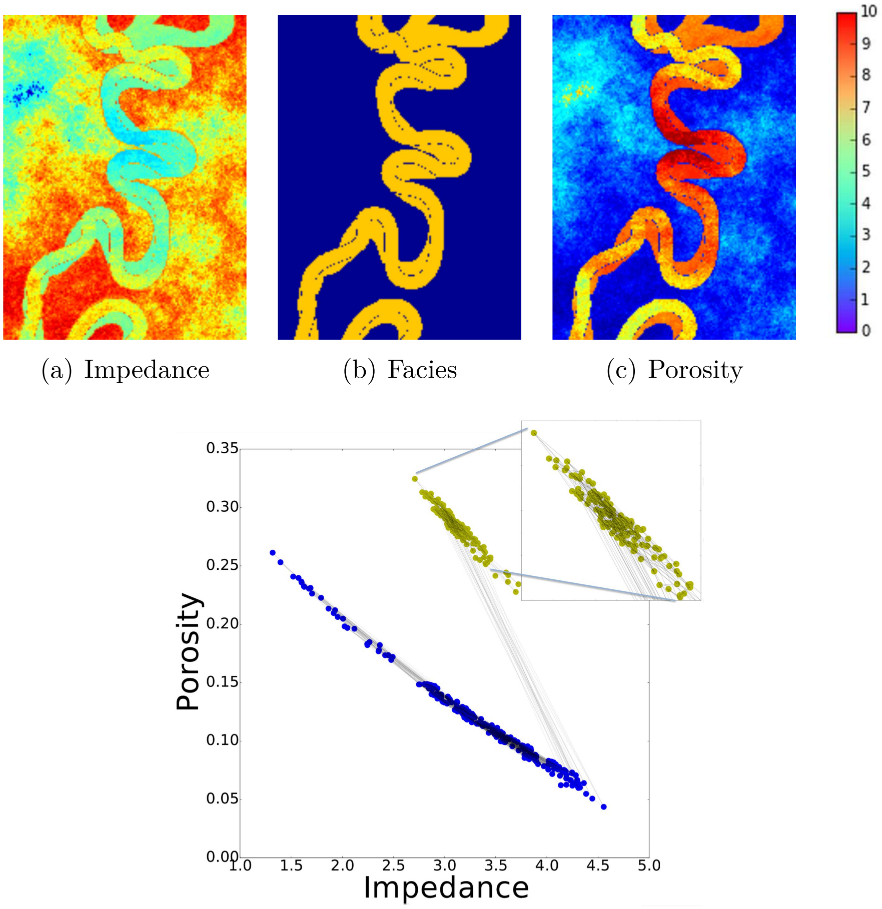
(c) Porosity

(d) Sand channels

Figure 1.4: Synthetic data reservoir example. From left to right: (a) seismic impedance input volume; (b) (latent) facies volume; (c) porosity output volume; (d) porosity output volume with applied transparency, showing only the high porosity sand channels [4].

It is reasonable to assume strong spatial structure connections within each facies. There are two facies in Fig. 1.4(b). The high porosity sand facies are colored yellow, while the low porosity shale facies are painted blue. Facies segmentation, nevertheless, is also an intricate task, due to the many complex geometric shapes that can co-exist in the reservoir. Besides, the usual methods applied in facies segmentation typically take into consideration only the seismic impedance inputs, which often contain overlaps for distinct facies.

The example in Figure 1.5 illustrates the porosity estimation problem to be solved. Fig. 1.5 (a) shows the seismic impedance from a horizontal slice in the synthetic reservoir. Fig. 1.5 (c) presents the corresponding porosity output to be estimated. Fig. 1.5 (b) reveals the hidden/latent facies. Looking at the impedance in (a) and referring to the latent facies in (b), one can clearly observe the overlap in the input values between the sand and shale facies, as similar impedance values (represented by colors) are present in both facies. This overlap can also be verified in the input (impedance) × target (porosity) crossplot in Fig. 1.5(d). The lines in this plot show the connections from each sample to its neighbors in a 4-tile lattice.



(a) Impedance (b) Facies (c) Porosity

(d) input vs. output

Figure 1.5: The porosity estimation problem. The goal is to estimate (c) porosity (unknown at most of the locations) from (a) impedance (known) by using a linear relationship between them. This relationship, however, depends on the (b) facies (unknown), and accurate facies estimation requires porosity measurements because of the overlapping marginal distribution of the impedance (d).

Since this problem deals with a combination of massive and imprecise unlabeled input instances with just a small fraction of precise regression labeled samples, it can be cast as a semi-supervised learning regression task [9], [10]. Moreover, because all the available data is already at hand, it is possible to adopt a *transductive setting* approach [11], [9], where labels of the unlabeled examples are estimated by learning a function defined only over the point cloud data.

Many current methods for porosity estimation rely upon standard geostatistical approaches, like in [12], [13], [14], [15], and [16], but the task still remains a challenge. Those methods are in general manual labor, time-consuming processes, demanding considerable expert knowledge during design parameterization. Chapter 2 describes the geostatistics fundamentals in more detail.

Prediction of porosity and other reservoir variables has also been addressed in several geophysics applications that, e.g., combine rock physics models with seismic inversion. Rock physics fundamentals are described in, e.g., [17], [18], [19], and [20]. Petrophysical seismic inversion formulations are depicted in [21], [22], [23], [24]. Gaussian mixture models for estimation of reservoir variables from seismic inversion and rock physics are presented in [25]. Lithology and fluid prediction classification based on Markov chain models are described in [26] and [16]. Also, joint inversion approaches for lithology and elastic properties have been proposed by [27], [18], among others. In this thesis, however, the focus is on automatic porosity estimation from already inverted seismic impedance volumes and sparse porosity samples located in a few exploratory wells, a typical problem faced by geologists during the evaluation of a reservoir in the exploration phase, where a rock physics model is not commonly available.

There are also machine learning and soft computing related approaches on porosity prediction [28], [29] and facies classification [30], [31], but none of them take into consideration the spatial structure in the reservoir.

Spatial structure has been modeled by conditional random fields (CRFs) [32], and their extensions comprise diverse continuous methods [33], [34], [35]. For kernel machines, the classical structured output support vector machines (SSVM) [36] allow to learn on joint feature maps (see Section A.2) and extensions to regression can be found in [37], [38], [39]. Those methods, however, require multiple, independent, fully-labeled samples, and are not directly applicable to this setting, where only dependent partially-labeled samples are provided.

Transductive Regression [40], [41] copes with the semi-supervised setting by inferring virtual labels for unlabeled examples by superposition of information of labeled examples [40]. Here, interactions between examples are imposed implicitly by choosing an appropriate metric. Those methods, however, do not take latent dependency structure into account.

Methods based on laplacian regularized learning machines [42], [43], [44] assume that data lies on a manifold in transductive or semi-supervised settings, but the computational complexity prevents their application to datasets with millions of samples.

Another line of research is a mixture of experts model [45], [46], [47], [48], where multiple regression models (experts) are trained, and one (or a weighted sum) of them is used to predict the output label of new samples.

Unfortunately, none of the aforementioned methods are able to simultaneously cope with the problem setting presented in this thesis, which can be summarized as follows:

- overlapping clusters in the input space;
- scarce labeled data;
- inference of spatial structures in latent space;
- regression based on inferred structured latent states;
- transductive setting.

1.2 Objective and Contributions

The main object of this work is to provide a new methodology for automatic porosity prediction in oil and gas reservoirs, given impedance cube volumes already available to the geologist from previous application of seismic inversion methods and sparse porosity samples located in a small group of exploratory wells.

To cope with the problem setting described in the previous section, the methodology contemplates the following requirements:

- transductive/semi-supervised learning;
- regression with continuous labels;
- Non-i.i.d. data with given dependency structure;
- latent state inference;
- mixture model.

A novel nonlinear method for porosity estimation is proposed, which takes into account geological facies as a latent variable with spatial dependency structure. It combines the concepts of Markov random fields, transductive learning, regression, and joint feature maps. The *Transductive Conditional*

Random Field Regression (TCRFR) method is able to infer the latent facies states in the reservoir by combining the local, limited and accurate porosity labeled data from the available wells with the plenty but imprecise unlabeled impedance information located everywhere in the volume. The corresponding porosities can be then estimated within each facies.

Moreover, two graph-based preprocessing techniques are introduced, mainly inspired in the image processing literature, which allow TCRFR to handle the extreme but realistic cases where the number of labeled samples in the reservoir correspond to less than 0.005% of all the available data. This workflow containing the preprocessing techniques plus TCRFR is defined as the *TCRFR Pipeline*.

Experiments are performed on both synthetic and real-world data reservoirs and show that the proposed TCRFR pipeline is able to successfully infer the correct geological facies geometric shapes and related porosities. The TCRFR accuracy performance is compared to other state-of-the-art baseline methods, as well as with the classic geostatistical approach.

1.3 Organization

The organization of this thesis is as follows.

Chapter 2 describes the fundamentals of geostatistics, the standard approach applied today for porosity estimation.

Chapter 3 explains the mathematical background behind conditional random fields (CRFs), the structured learning basis method used in the new proposed model.

Chapter 4 presents in detail the proposed Transductive Conditional Random Field Regression model (TCRFR).

Chapter 5 proposes the TCRFR Pipeline to handle realistic porosity prediction problem settings, where only a scarce amount of labeled data is available from a few wells located in the reservoir.

Chapter 6 presents the results and discussions for the application of the TCRFR method and the TCRFR Pipeline to both synthetic and real reservoir data sets, comparing their performance with other machine learning state-of-the-art methods and also with the classic geostatistical approach.

Finally, Chapter 7 concludes the thesis and proposes future directions.

2
Geostatistics

Spatial interpolation has application in many and diversified fields such as geology, geophysics, meteorology, medicine, engineering, economy, and social sciences. It consists basically in estimating the value z of a random variable $Z(\boldsymbol{u}_0)$ at any coordinate $\boldsymbol{u}_0 = (x_0, y_0, z_0)$, given an input set of sampled data points $z(\boldsymbol{u}_\alpha)$ located at coordinates $\boldsymbol{u}_\alpha = (x_\alpha, y_\alpha, z_\alpha)$ that usually have a non-uniform distribution [49].

Geostatistics [50] is today the main approach applied to spatial interpolation related to physical phenomena and, in particular, petrophysical properties estimation. It consists on the application of statistical methods that take into account the spatial correlations between the random variables. The following sections describe the main concepts behind the geostatistics framework. Most of the contents in this chapter are summarized from [13], [51], and [52].

2.1
Random variables and stationarity

Formally, any property value spatially located at position $\boldsymbol{u}_1$ is interpreted as the $z(\boldsymbol{u}_1)$ realization of the random variable $Z(\boldsymbol{u}_1)$. So, in the space $\mathcal{S}$ where all the samples are located, we have the realizations of N correlated random variables $Z(\boldsymbol{u}_1), Z(\boldsymbol{u}_2), ..., Z(\boldsymbol{u}_N)$. The set of correlated random variables is a *random function* where only one realization $z(\boldsymbol{u}_\alpha)$ for each random variable is known, that is, the sampled data set.

With just one realization for each random variable, it is practically impossible to determine statistical parameters for the individual variables or for the random function. To work around this problem, two criteria are considered.

The first one assumes *mean stationarity*, i.e., all random variables have the same mean,

$$E\{Z(\boldsymbol{u}_1)\} = E\{Z(\boldsymbol{u}_2)\} = \cdots = E\{Z(\boldsymbol{u}_N)\} = m \tag{2-1}$$

With this assumption, m becomes independent of the spatial location

$(\boldsymbol{u}_i)$ and can be estimated by simple arithmetic mean:

$$m = \frac{1}{N}\sum_{\alpha=1}^{N} Z(\boldsymbol{u}_\alpha)$$

The second criterion assumes *variance stationarity.* Applying the same reasoning, we get

$$E\{[Z(\boldsymbol{u}_1) - m]^2\} = E\{[Z(\boldsymbol{u}_2) - m]^2\} = \cdots = E\{[Z(\boldsymbol{u}_N) - m]^2\} = \sigma^2 \quad (2\text{-}2)$$

These decisions of stationarity result in that the covariance between any two variables separated by a distance vector $\boldsymbol{h}$ depends only on $\boldsymbol{h}$:

$$C(Z(\boldsymbol{u}_i), Z(\boldsymbol{u}_j)) = C(Z(\boldsymbol{u}_i), Z(\boldsymbol{u}_i + \boldsymbol{h})) = C(\boldsymbol{h}), \quad (2\text{-}3)$$

or, in other words,

$$C(\boldsymbol{h}) = E\{Z(\boldsymbol{u}_i + \boldsymbol{h})Z(\boldsymbol{u}_i)\} - [E\{Z(\boldsymbol{u}_i)\}]^2 \quad (2\text{-}4)$$

At $\boldsymbol{h} = 0$ we get the stationary variance σ^2:

$$\begin{aligned} C(0) &= E\{Z(\boldsymbol{u}_i + 0)Z(\boldsymbol{u}_\alpha)\} - [E\{Z(\boldsymbol{u}_\alpha)\}]^2 \\ &= E\{Z(\boldsymbol{u}_\alpha)^2\} - [E\{Z(\boldsymbol{u}_\alpha)\}]^2 \\ &= Var(Z(\boldsymbol{u}_\alpha)) = \sigma^2 \end{aligned} \quad (2\text{-}5)$$

Those are strong assumptions and, in practice, their degree of applicability depends on the sample homogeneity of the random variable distribution in the $\mathcal{S}$ space. In most cases, a local neighborhood is defined around each point being estimated, restricting the stationarity to a subset of $\mathcal{S}$.

2.2 Experimental variograms

From Equations (2-1), (2-2), and (2-4) it is possible to define a standardized stationary correlogram

$$\rho(\boldsymbol{h}) = \frac{C(\boldsymbol{h})}{C(0)}, \quad (2\text{-}6)$$

and also a second order moment, known as the *variogram*:

$$2\gamma(\boldsymbol{h}) = E\{[Z(\boldsymbol{u}_\alpha + \boldsymbol{h}) - Z(\boldsymbol{u}_\alpha)]^2\} \quad (2\text{-}7)$$

The *semivariogram* is defined as one half of the variogram, or $\gamma(\boldsymbol{h})$. The semivariogram encodes data about spatial variance over a region at a given distance or lag. Points that are spatially close should share similar features and points that are separated by greater distances should have less correlation. So, the semivariogram allows modeling the similarity points in a field as a function of distance.

One can observe that, assuming stationarity of mean and variance, the covariance, the correlogram, and the variogram are equivalent forms to define two-point correlations:

$$C(\boldsymbol{h}) = C(0) \times \rho(\boldsymbol{h}) = C(0) - \gamma(\boldsymbol{h})$$

Using the available experimental data set and Equation (2-7), the *experimental semivariogram* is defined as

$$\hat{\gamma}(\boldsymbol{h}) = \frac{1}{2N(\boldsymbol{h})} \sum_{N(\boldsymbol{h})} [z(\boldsymbol{u}_\alpha + \boldsymbol{h}) - z(\boldsymbol{u}_\alpha)]^2, \qquad (2\text{-}8)$$

where $N(\boldsymbol{h})$ represents the number of points $\boldsymbol{u}_\alpha$ separated by distance $\boldsymbol{h}$.

The distance vector $\boldsymbol{h}$ is specified with a direction and a distance tolerance known as *lag*. Both the direction and the lag must be defined by the user. The defined direction usually reflects the main directions of continuity (i.e., major, minor, and intermediate) in the reservoir.

Let us illustrate the concepts described so far through an example extracted from [5]. Here we are given the region in Figure 2.1 with 80 porosity sample points. The goal is to estimate the porosity for the whole region using those 80 sample points as input. Generally in surveys it is common to specify one point in latitude and longitude and then measure all other locations as North and East of that point, hence the Northing and Easting in the plot.

Applying Equation (2-8), with a direction and a lag defined by the user results in the experimental semivariogram shown in Figure 2.2.

Let's analyze how this plot is created. It describes how the variance (vertical axis) changes as a function of lag increments (horizontal axis, in meters). The lag distance defined by the user in this case is 500 meters. There are 500 meter increments from 0 to 10,000 meters. The blue dots on the curve represent the 500 meter increments. All the pairs of points in Figure 2.1 are obtained for each lag interval where the distance is in the lag range, e.g., all pairs within distance from 0 to 499 meters, then all pairs within distance from 500 to 1,000 meters, and so on. Within each lag the actual $\boldsymbol{h}$ distance is calculated between each pair and then Equation 2-8 is applied for all the

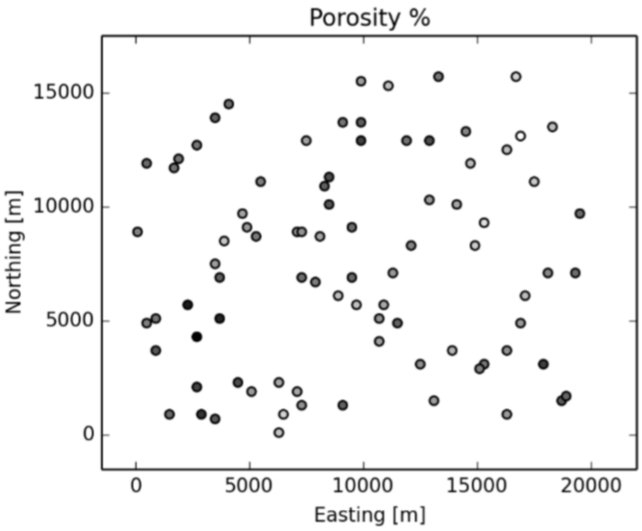


Figure 2.1: An example of a porosity estimation problem setting. The points refer to porosity sample values, ranging from low (white dots) to high porosity (black dots) [5].

points within the lag. In other words, the variogram is only calculated at the specific lag distances.

As expected, it can be seen that the variance increases (or, in other words, the covariance decreases) with the increasing distance to a point where there is no more correlation between the points and the semivariance value reflects just the stationary variance σ^2.

In this simple example, the variogram construction is relatively easy. It is important, nevertheless, to keep in mind that in most real cases variograms are anisotropic, i.e., geologic continuity and, consequently, the variogram continuity are direction dependent. In sedimentary structures, continuity is more evident in the horizontal direction than in the vertical direction [51]. The horizontal continuity depends on the direction of the sedimentary deposition and these directions need to be geologically interpreted by an expert. For the definition of the scalar distance h, three angles define the orthogonal x, y and z coordinates and then the components of the distance vectors are scaled by the three scalar parameters s_x, s_y and s_z, which depend on the interpreted continuity in each direction:

$$h = \sqrt{(h_x/s_x)^2 + (h_y/s_y)^2 + (h_z/s_z)^2}$$

In this equation, h_x, h_y and h_z are the components of vector $\boldsymbol{h}$.

The example in Figure 2.3 illustrates how the direction of $\boldsymbol{h}$ can change. It shows an xy porosity map and the blue arrows show the (h_x, h_y) variogram

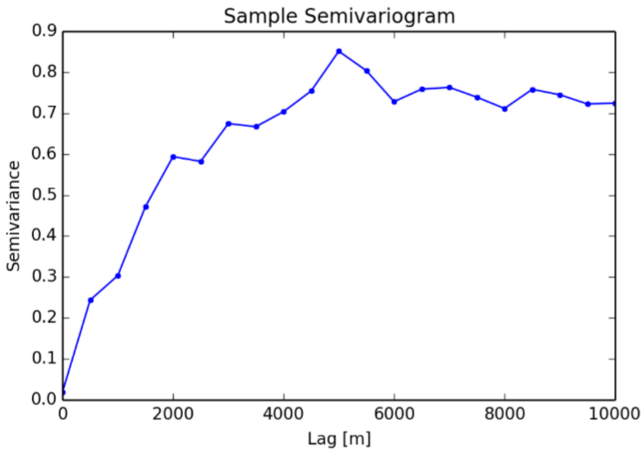


Figure 2.2: The experimental semivariogram [5].

major directions of $\boldsymbol{h}$ in different locations.

2.3 Variogram modeling

The experimental semivariogram points defined in the previous section are not used directly. Instead, a parametric variogram model needs to be chosen by the user and fitted to the experimental data points, to be later used by the interpolation function. These models are positive definite functions and the most common are the spherical, exponential, and gaussian models. For more information on the different features of each variogram model, please refer to [51].

In the current example, a spherical model is fit to the data and the generated model is represented by the green line shown in Figure 2.4.

2.4 Kriging

From the previously defined semivariogram model, it is now necessary to interpolate the sampled data points and create a map of the estimated porosity.

Considering $Z(\boldsymbol{u}_0)$ a random variable located at $\boldsymbol{u}_0$ and the known values of the neighbor sample data points as $z(\boldsymbol{u}_\alpha), \alpha = 1, 2, ..., N$, it is established by stationarity that

$$E\{Z(\boldsymbol{u}_\alpha)\} = E\{Z(\boldsymbol{u}_0)\} = m$$

Because of the second order stationarity, it is also know that the variogram and the covariance only depend on $\boldsymbol{h}$.

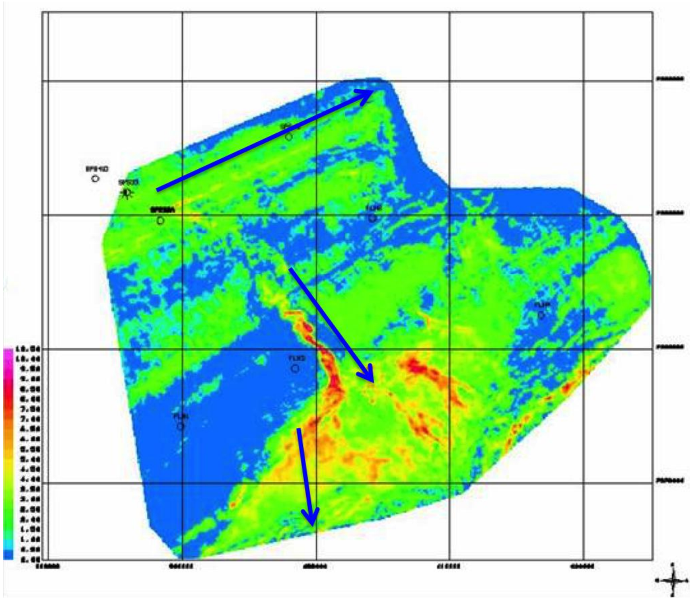


Figure 2.3: Variogram direction dependency in sedimentary structures.

The estimator $[Z(\boldsymbol{u}_0)]^* = \sum_{\alpha=1}^{N} \lambda_\alpha Z(\boldsymbol{u}_\alpha)$ can be considered as a random variable located at $\boldsymbol{u}_0$ resulting from the linear combination of $Z(\boldsymbol{u}_\alpha), \alpha = 1, 2, ..., N$.

Defining $\epsilon(\boldsymbol{u}_0)$ as the error of the $[Z(\boldsymbol{u}_0)]^*$ estimator, we have

$$\epsilon(\boldsymbol{u}_0) = [Z(\boldsymbol{u}_0)]^* - [Z(\boldsymbol{u}_0)] = \sum_{\alpha} \lambda_\alpha Z(\boldsymbol{u}_\alpha) - Z(\boldsymbol{u}_0)$$

From Equation (2-1),

$$E\{\epsilon(\boldsymbol{u}_0)\} = E\left\{\sum_{\alpha} \lambda_\alpha Z(\boldsymbol{u}_\alpha)\right\} - E\{Z(\boldsymbol{u}_0)\} = 0$$

$$E\left\{\sum_{\alpha} \lambda_\alpha Z(\boldsymbol{u}_\alpha)\right\} = E\{Z(\boldsymbol{u}_0)\} \tag{2-9}$$

To avoid bias in the estimation, it is necessary to make

$$\sum_{\alpha} \lambda_\alpha = 1$$

Besides the minimum error $\epsilon(\boldsymbol{u}_0)$, minimum variance is also necessary as two estimators can have the $E\{\epsilon(\boldsymbol{u}_0)\} = 0$, but one of them can have a lower

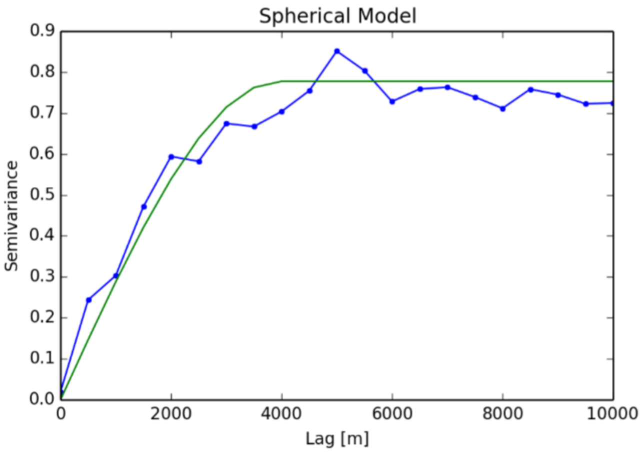


Figure 2.4: The semivariogram experimental (blue) and model (green) curves [5].

dispersion, making it a better estimator:

$$var\{\epsilon(\boldsymbol{u}_0)\} = var\{[Z(\boldsymbol{u}_0)]^* - Z(\boldsymbol{u}_0)\} = E\left\{ \left[\sum_{\alpha} \lambda_\alpha Z(\boldsymbol{u}_\alpha) - Z(\boldsymbol{u}_0) \right]^2 \right\}$$

Decomposing the previous expression, the variance becomes

$$\begin{aligned} var\{\epsilon(\boldsymbol{u}_0)\} &= E\left\{ \sum_{\alpha}\sum_{\beta} \lambda_\alpha \lambda_\beta Z(\boldsymbol{u}_\alpha) Z(\boldsymbol{u}_\beta) \right\} + E\{Z(\boldsymbol{u}_0)^2\} - 2E\left\{ \sum_{\alpha} \lambda_\alpha Z(\boldsymbol{u}_\alpha) Z(\boldsymbol{u}_0) \right\} \\ &= \sum_{\alpha}\sum_{\beta} \lambda_\alpha \lambda_\beta E\{Z(\boldsymbol{u}_\alpha) Z(\boldsymbol{u}_\beta)\} + E\{Z(\boldsymbol{u}_0)^2\} - 2\sum_{\alpha} E\{Z(\lambda_\alpha \boldsymbol{u}_\alpha) Z(\boldsymbol{u}_0)\} \end{aligned}$$

Once the covariance model or the variogram has been defined, the previous variance expression can be rewritten as a function of the covariances between the sampled data points and the covariances between the sampled data points and the point to be estimated:

$$var\{\epsilon(\boldsymbol{u}_0)\} = C(0) + \sum_{\alpha}\sum_{\beta} \lambda_\alpha \lambda_\beta C(\boldsymbol{u}_\alpha, \boldsymbol{u}_\beta) - 2\sum_{\alpha} \lambda_\alpha C(\boldsymbol{u}_\alpha, \boldsymbol{u}_0) \qquad (2\text{-}10)$$

The linear estimation method used in geostatistics, known as *Kriging*, is defined at this point.

Kriging comprehends a family of interpolation methods which are generalized forms of univariate and multivariate linear regression models for estimation at a point location over an area or volume. The interpolated values are modeled by a Gaussian process governed by prior covariances [53]. They are linear-weighted averaging methods, similar to other interpolation methods, however their weights depend not only on distance, but also on the direction and orientation of the neighboring data provided by the previously defined variograms. These methods honor the measurements of the sampled data points $Z(\boldsymbol{u}_\alpha), \alpha = 1, 2, ..., N$, keeping them fixed and limiting in this way the smoothness in the estimated results.

The basic algorithm, called Simple Kriging, is a linear combination of a set of N sampled variables $Z(\boldsymbol{u}_\alpha), \alpha = 1, 2, ..., N$ that are neighbors of an unknown sample located at $\boldsymbol{u}_0$ and that satisfies the conditions described previously, i.e., $E\{\epsilon(\boldsymbol{u}_0)\} = 0$, $\min\{var(\epsilon(\boldsymbol{u}_0))\}$, and $E\{Z(\boldsymbol{u}_1)\} = E\{Z(\boldsymbol{u}_2)\} = \cdots = E\{Z(\boldsymbol{u}_N)\} = m$.

Depending on the stochastic relations of the random variables and the degree of the assumed stationarity, different kriging methods can be applied for calculating the λ_α weight parameters. Some classical methods are Simple Kriging, Ordinary Kriging, Universal Kriging, and Indicator Kriging. For more information, please refer to [13, 51].

Equation (2-10) is now optimized to obtain the weights that minimize the estimation variance. This is accomplished by applying partial derivatives of Equation (2-10) with respect to each weight λ_α:

$$\frac{\partial[\sigma^2(\boldsymbol{u}_0)]}{\partial\lambda_\alpha} = 2\sum_{\beta=1}^{N} \lambda_\beta C(\boldsymbol{u}_\beta - \boldsymbol{u}_\alpha) - 2C(\boldsymbol{u}_0 - \boldsymbol{u}_\alpha),\ \alpha = 1, ..., N$$

Setting the previous equation to zero, the minimum weights λ_α are calculated as

$$\sum_{\beta=1}^{N} \lambda_\beta C(\boldsymbol{u}_\beta - \boldsymbol{u}_\alpha) = C(\boldsymbol{u}_0 - \boldsymbol{u}_\alpha) \tag{2-11}$$

For three sample points, for instance, the following system of equations (known as simple kriging system) would be defined:

$$\begin{aligned}
C(\boldsymbol{u}_1 - \boldsymbol{u}_1) \times \lambda_1 + C(\boldsymbol{u}_1 - \boldsymbol{u}_2) \times \lambda_2 + C(\boldsymbol{u}_1 - \boldsymbol{u}_3) \times \lambda_3 = C(\boldsymbol{u}_0 - \boldsymbol{u}_1)\\
C(\boldsymbol{u}_2 - \boldsymbol{u}_1) \times \lambda_1 + C(\boldsymbol{u}_2 - \boldsymbol{u}_2) \times \lambda_2 + C(\boldsymbol{u}_2 - \boldsymbol{u}_3) \times \lambda_3 = C(\boldsymbol{u}_0 - \boldsymbol{u}_2)\\
C(\boldsymbol{u}_3 - \boldsymbol{u}_1) \times \lambda_1 + C(\boldsymbol{u}_3 - \boldsymbol{u}_2) \times \lambda_2 + C(\boldsymbol{u}_3 - \boldsymbol{u}_3) \times \lambda_3 = C(\boldsymbol{u}_0 - \boldsymbol{u}_3)
\end{aligned}$$

or, in matrix notation:

$$\begin{bmatrix} C(\boldsymbol{u}_1 - \boldsymbol{u}_1) & C(\boldsymbol{u}_1 - \boldsymbol{u}_2) & C(\boldsymbol{u}_1 - \boldsymbol{u}_3) \\ C(\boldsymbol{u}_2 - \boldsymbol{u}_1) & C(\boldsymbol{u}_2 - \boldsymbol{u}_2) & C(\boldsymbol{u}_2 - \boldsymbol{u}_3) \\ C(\boldsymbol{u}_3 - \boldsymbol{u}_1) & C(\boldsymbol{u}_3 - \boldsymbol{u}_2) & C(\boldsymbol{u}_3 - \boldsymbol{u}_3) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} C(\boldsymbol{u}_0 - \boldsymbol{u}_1) \\ C(\boldsymbol{u}_0 - \boldsymbol{u}_2) \\ C(\boldsymbol{u}_0 - \boldsymbol{u}_3) \end{bmatrix}$$

In general,

$$C\lambda = c \Rightarrow \lambda = C^{-1}c \tag{2-12}$$

Back to the current example, in Equation (2-12) C is the matrix of covariances calculated using the spherical model, λ_α is the vector of weights, and c is the vector of covariances between the data points and an unsampled point. For this case Simple Kriging was chosen. The kriging function uses in the calculation the sampled data points $Z(\boldsymbol{u}_\alpha), \alpha = 1, 2, ..., N$, the model, the distances $\boldsymbol{h}$, the coordinates of the unsampled point $\boldsymbol{u}_0$, and the number of surrounding neighbors points nbg to the unsampled point. It is important to keep in mind that this system of equations needs to be solved for each unsampled point in the region.

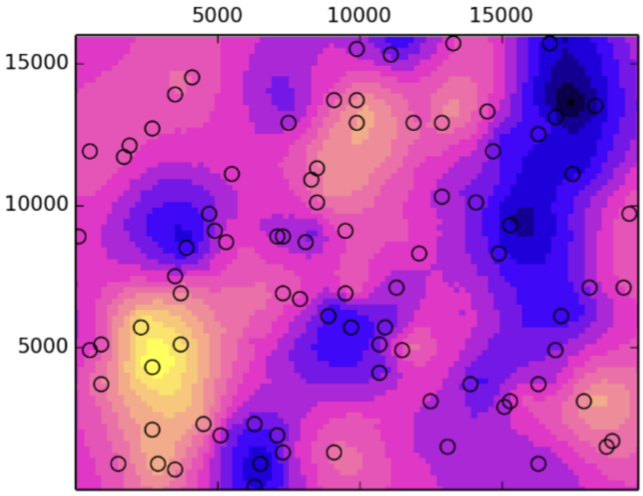


Figure 2.5: Estimated porosity from kriging interpolation. Porosity ranges from low (in blue) to high (in yellow) [5].

The porosity estimation result from the application of Simple Kriging is presented in Figure 2.5.

2.5 Secondary variables

In many cases the number of available samples is not sufficient for a good interpolation. Porosity estimation during the exploration stage is a good example. Usually, only a small number of wells exist in the reservoir, so just a few number of samples are available. In these cases, it is common to use a secondary variable to help the estimations.

Seismic impedance is usually negatively correlated with porosity and it is normally used as a secondary variable in porosity estimation. Due to geological complexities and the seismic data vertical low resolution as a result of limitations in the acquisition process, seismic imprecisely measures the average porosity at each point in the reservoir. Typically, the seismic vertical data resolution in a reservoir volume is 10 to 100 times lower than the well resolution, while the areal (xy) resolution is usually comparable [51]. The degree of correlation must be calibrated for each reservoir. After this calibration, the seismic impedance information can be used in a similar way as the well information.

The most common methods for applying kriging with secondary variables are kriging with varying local means, kriging with external drift, and collocated cokriging. For more information, please refer to [13, 54].

In theory, geostatistical models have been devised for any number of secondary variables [51], however, they are difficult to apply for more than one in practice. This difficulty arises from the need of modeling and inferring the cross-covariance matrix between the input variables involved. Because of implementation problems [55], joint probability simulations are rarely used in its full version and simplifications are adopted, including Markov models and factorization methods.

3
Conditional Random Fields

One vital requirement for many real world problems nowadays is the ability to classify multiple variables that are dependent on each other [56] in spatial or temporal structures. The fields of application include object classification in an image [57], natural language processing [58], and segmenting DNA sequences [59]. In such applications, one wants to predict structured data encoded on a label output vector $\boldsymbol{y} = (y_0, y_1, ..., y_n) \in \mathcal{Y}^n$ of random variables given an observed feature vector $\boldsymbol{x} = (\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_n) \in \mathcal{X}^{n \times D}$. Each $\boldsymbol{x}_i$ can be, for instance, an input image and y_i, the corresponding label for this image. In those cases, there is a mapping $h : \mathcal{X} \to \mathcal{Y}$ to be learned, so that

$$h(\boldsymbol{x}) = \operatorname*{argmax}_{y \in \mathcal{Y}} f(\boldsymbol{x}, y),$$

where f is a compatibility function that expresses how well y fits the input $\boldsymbol{x}$.

This *structured learning* scenario, where multiple and interdependent class and observation variables are considered, implies a complex probability distribution. *Probabilistic graphical models* map these probability distributions in a graphical form. Conditional random fields (CRFs) [32] are probabilistic graphical models used for labeling and segmenting structured data, such as sequences, trees, and grids [60].

The following sections describe graphical representations in general and, in particular, conditional random fields. These discussions are adapted from [56] and [61].

3.1
Graphical representation

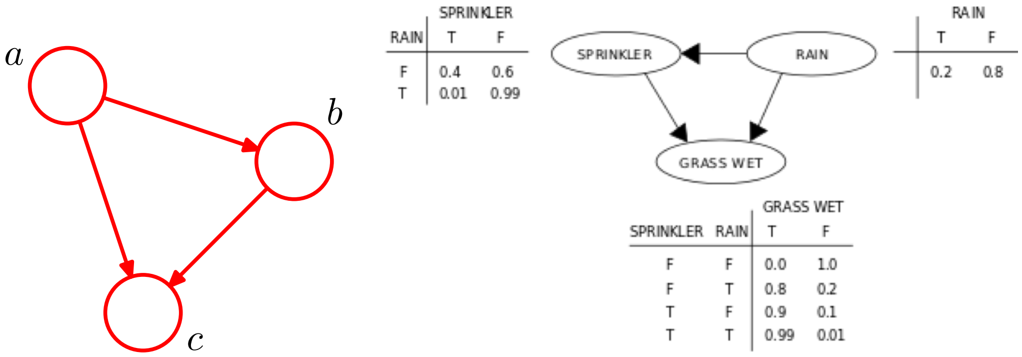
A probabilistic graphical model consists on a diagrammatic representation of a probability distribution. In this graph $G = (V, E)$, comprising a set V of vertexes and a set E of edges, there is a node for each random variable. The absence of an edge between two nodes means that the corresponding random variables are conditionally independent from each other, given a third random variable. Conditional independence between two random variables a and b given some other random variable c means that they are independent in their conditional probability distribution, i.e., $p(a, b|c) = p(a|b, c)p(b|c) = p(a|c)p(b|c)$.

Conditional independence is an important concept as it can be used to decompose complex probability distributions into a product of factors, each consisting of the subset of corresponding random variables. As a result, complex computation for learning and inference algorithms can be decomposed, or *factored*, in a much more efficient way (more on factors in the Appendix, Section A.1). Denoting by $\boldsymbol{x}$ the values of all unobserved variables in a graph, the factorization of a probability distribution is written as a product of factors Ψ_s, with $\boldsymbol{v}_s$ representing the subset of the respective random variables constituting such a factor and s as the factor subset indexes:

$$p(\boldsymbol{x}) = \prod_s \Psi_s(\boldsymbol{v}_s) \tag{3-1}$$

There are two types of graphical models, directed and undirected. In directed graphical models, also known as Bayesian Networks, the random variables and their conditional dependencies are represented as directed acyclic graphs (DAG). Figure 3.1(a) [6] shows an example of a directed graphical model. In this particular example, the joint probability distribution of the random variables is given by the factorization

$$p(a, b, c) = p(c|a, b)p(b|a)p(a) \tag{3-2}$$



	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99

RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

(a) Directed graph

(b) Causal connections

Figure 3.1: An example of directed graphical model [6].

Directed graph models are used in cases where there are influence connections among the random variables, as show in Figure 3.1(b). In this case, *grass wet* depends on the probability of the sprinkler being *on* or *off* and on the chance of raining. *Sprinkler*, in turn, also depends on the probability of raining. Directed graph models always have a direct probabilistic interpretation.

The remaining of this thesis focus on the second type of graphical models, discussed in the next section.

3.1.1 Undirected Graphical Models

Undirected graphical models differ from directed graph models in that they can be cyclic and they do not necessarily have a direct probabilistic interpretation, as explained in latter paragraphs. They are also known as Markov Random Fields (MRF) because the associated random variables satisfy the global Markov property [62].

Let $G = (V, E)$ be a graph with vertexes $v \in V$ and edges $e \in E$. In this graph, vertexes $V = X \cup Y$, with X and Y as two sets of random variables. Set X represents the observed input variables and set Y represents the output variables. For the following discussion, it is important to define the graphical concept known as *clique*. A clique is defined as a subset of the vertexes in the graph G such that there exists a link between all pairs of nodes in this subset. In other words, the set of nodes in a clique is fully connected. A *maximal clique* is a clique such that it is not possible to include any other nodes from the graph in the subset without it ceasing to be a clique. Figure 3.1 [6] illustrates these concepts. There we have a four-node undirected graph showing a clique, outlined in green, and a maximal clique, outlined in blue.

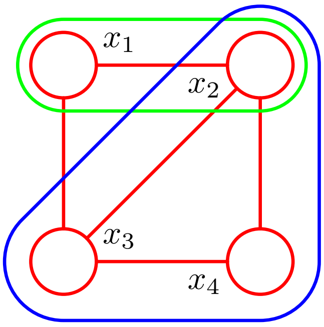


Figure 3.2: An example of a four-node undirected graph showing a clique in green and a maximal clique in blue [6].

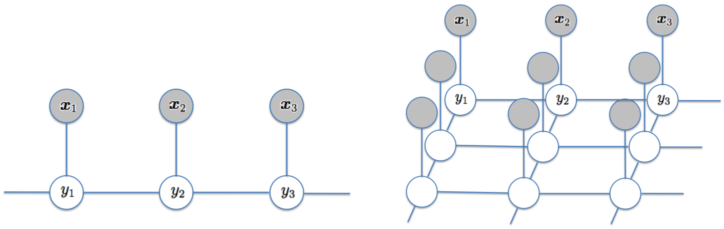
A probability distribution can be represented by an undirected graphical model using a product of non-negative functions in the set of the maximal cliques $\mathcal{C}$ of graph G. The factorization is performed in a way that conditionally independent nodes do not appear within the same factor, which means that they belong to different cliques:

$$p(\boldsymbol{x}) = \frac{1}{Z}\prod_{C \in \mathcal{C}} \Psi_C(\boldsymbol{x}_C) \tag{3-3}$$

Equation (3-3) reflects the Hammersley-Clifford theorem [63], the fundamental theorem of random fields, which stipulates that the probability of a particular set of values on the random variables on an undirected graphical model is a product of the potential functions over cliques of the graph. The factors $\Psi_C \geq 0$ are known as *potential functions* (also *feature* or *compatibility* functions) of the random variables $\boldsymbol{x}_C$ within a clique $C \in \mathcal{C}$. The potential functions may be any arbitrary function and do not necessarily have to be probability functions, i.e., they do not need to sum up to 1. This is a contrast to directed graphs where the joint distribution is factorized into a product of conditional distributions. As a consequence, normalization of the product of potential functions is necessary to achieve a proper probability measure. This is carried out by a normalization factor Z:

$$Z = \sum_{\boldsymbol{x}}\prod_{C \in \mathcal{C}} \Psi_C(\boldsymbol{x}_C) \tag{3-4}$$

Figure 3.3 shows two MRF examples:



(a) MRF sequence (b) MRF grid

Figure 3.3: Two examples of undirected graphical models.

Fig. 3.3(a) represents a chain or sequence. The $\boldsymbol{x}'s$ are the observable variables and the $y's$ represent the labels. The $\boldsymbol{x}$ variables could represent the four amino acids A, C, G, T, and the y variables would be the classification of a DNA sequence. The joint distribution is

$$P(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, y_1, y_2, y_3) = \frac{1}{Z}\Psi_{12}(y_1, y_2)\Psi_{23}(y_2, y_3)\Psi_1(\boldsymbol{x}_1, y_1)\Psi_2(\boldsymbol{x}_2, y_2)\Psi_3(\boldsymbol{x}_3, y_3)$$

Fig. 3.3(b) shows a grid. The $\boldsymbol{x}$ variables could be RGB pixel intensities in an image and the y variables would be the output classification, like *person*, *grass*, *sky*, etc. The joint distribution in this case is

$$P(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, y_1, y_2, y_3) = \frac{1}{Z} \prod_i \Psi_i(\boldsymbol{x}_i, y_i) \prod_{(i,j) \in E} \Psi_{ij}(y_i, y_j)$$

It is worth mentioning that in both examples the structured relationships occur in the output space. The input variables are completely independent from one another. It is also possible to observe the Markov property. For instance, in Fig. 3.3(a), conditioned on y_2, y_1 is independent of y_3.

The potential functions are non-negative (i.e. probabilities) and expressed as exponentials, so that

$$\Psi_C(\boldsymbol{x}_C) = \exp\{-E(\boldsymbol{x}_C)\}, \tag{3-5}$$

where $E(\boldsymbol{x}_C)$ is known as the *energy function*. The joint distribution is defined as the product of potentials [63] and so the total energy is obtained by adding the energies of each of the maximal cliques [6]. The following example for some 4-node MRF illustrates this concept:

$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z} f_1(x_1, x_2) f_2(x_2) f_3(x_2, x_3, x_4) \tag{3-6}$$

$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z} e^{\ln(f_1(x_1,x_2)) + \ln(f_2(x_2)) + \ln(f_3(x_2,x_3,x_4))} \tag{3-7}$$

Equation (3-6) is in the probability domain, while Equation (3-7) is in the energy or *log probability* domain. For computational reasons, e.g., real numbers approximation and efficiency, it is usually preferable to represent probabilities in the logarithm space.

The decomposition of a graph as a product of factors over subsets of variables can be expressed explicitly in a *factor graph* representation. Given a distribution that is expressed in terms of an undirected graph, it is straightforward to convert it to a factor graph. To do this, one creates variable nodes corresponding to the nodes in the original undirected graph, and then creates additional factor nodes corresponding to the maximal cliques x_s. The factors $f_s(x_s)$ are then set equal to the clique potentials. There may be several different factor graphs that correspond to the same undirected graph, with totally arbitrary functions, again making it necessary to calculate the normalization factor Z. The factor graph to associate to an undirected graphical model is going to depend on the specifics of problem setting, like functions or features connecting two or more variables. Figure 3.4 [6] illustrates this concept. Fig. 3.4(a) shows a three-node undirected graph with a single clique potential $\Psi(x_1, x_2, x_3)$; in (b) we see a factor graph with factor $f(x_1, x_2, x_3) = \Psi(x_1, x_2, x_3)$; in (c) another factor graph for the same distribution, with $\Psi(x_1, x_2, x_3) = f_a(x_1, x_2) f_b(x_1, x_3) f_c(x_2, x_3)$; and in (d) still another factor graph where $\Psi(x_1, x_2, x_3) = f_a(x_1, x_2, x_3) f_b(x_2, x_3)$.

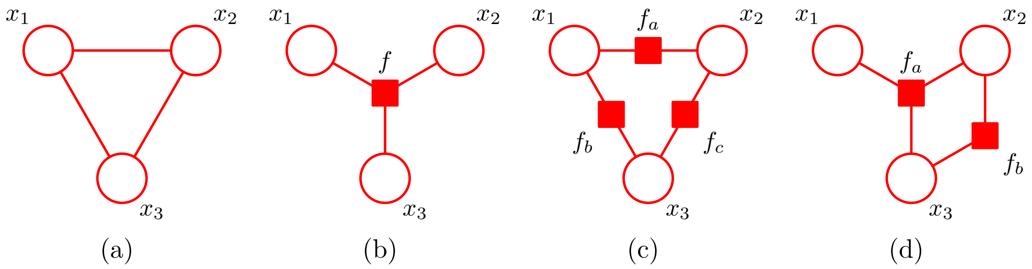


Figure 3.4: An undirected graph and three examples of factorization [6].

3.2 Conditional Random Fields

A conditional random field (CRF) [32] is a discriminative graphical probabilistic model that can be arbitrarily spatially or temporally structured. *Discriminative models* define a class of models used in machine learning for modeling the dependence of an unobserved variable y conditioned on an observed variable $\boldsymbol{x}$. Within a probabilistic framework, this is accomplished by modeling the conditional probability distribution $P(y|\boldsymbol{x})$ [56, 61]. In other words, discriminative models describe directly how to take a feature $\boldsymbol{x}$ and assign it a label y. Compared to generative models, which model the joint probability distribution $p(\boldsymbol{x}, \boldsymbol{y})$, discriminative models have the advantage of not requiring the knowledge of $p(\boldsymbol{x})$ distribution nor if the input features $\boldsymbol{x}$ are correlated or not.

Formally, CRFs compute the probability $p(\boldsymbol{y}|\boldsymbol{x})$ of a possible output $\boldsymbol{y} = (y_1, ..., y_n) \in \mathcal{Y}^n$ given the input observation $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \in \mathcal{X}^{n \times D}$, where D corresponds to the number of dimensions (or features) of $\boldsymbol{x}$. The conditional random field formulation can be derived from equation 3-3:

$$p(\boldsymbol{x}_C, \boldsymbol{y}_C) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \Psi_C(\boldsymbol{x}_C, \boldsymbol{y}_C)$$

The conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$ is defined as

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x}) &= \frac{p(\boldsymbol{y},\boldsymbol{x})}{p(\boldsymbol{x})} \\ &= \frac{p(\boldsymbol{y},\boldsymbol{x})}{\sum_{\boldsymbol{y}'} p(\boldsymbol{y}',\boldsymbol{x})} \\ &= \frac{\frac{1}{Z}\prod_{C\in\boldsymbol{C}} \Psi_C(\boldsymbol{x}_C,\boldsymbol{y}_C)}{\frac{1}{Z}\sum_{\boldsymbol{y}'}\prod_{C\in\boldsymbol{C}} \Psi_C(\boldsymbol{x}_C,\boldsymbol{y}'_C)} \end{aligned}$$

From the previous expression, the general formulation of conditional random fields is derived as

$$\boxed{p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\prod_{C\in\mathcal{C}} \Psi_C(\boldsymbol{x}_C,\boldsymbol{y}_C)} \tag{3-8}$$

As stated previously, Ψ_C are the different factors corresponding to maximal cliques in the graph G. Each factor is a potential function that combines different features f_i of the input observations and outputs.

The normalization factor $Z(\boldsymbol{x})$, also referred as the *partition function*, corresponds to the denominator of equation 3-8 and it is summed over all $y's$ to provide a proper conditional probability:

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}'}\prod_{C\in\mathcal{C}} \Psi_C(\boldsymbol{x}_C,\boldsymbol{y}'_C) \tag{3-9}$$

As already explained in Section 3.1.1, Equation (3-8) can be expressed as an energy function:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\exp\textstyle\sum_{C\in\mathcal{C}}(-E(\boldsymbol{x}_C,\boldsymbol{y}_C))$$

This negative energy function is usually written as a weighted sum of K real-valued potential functions $f_k(\boldsymbol{x}_C,\boldsymbol{y}_C)$, $f:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}$:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\exp\textstyle\sum_{C\in\mathcal{C}}\sum_{k=1}^{K}\lambda_k f_k(\boldsymbol{x}_C,\boldsymbol{y}_C) \tag{3-10}$$

The corresponding partition function is then

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}'}\exp\textstyle\sum_{C\in\mathcal{C}}\sum_{k=1}^{K}\lambda_k f_k(\boldsymbol{x}_C,\boldsymbol{y}'_C) \tag{3-11}$$

To prevent a significant growth in the number of parameters proportional to the number of nodes and edges in the graph, with an specific λ_k for each potential function, it is common practice to group and share sets of factors and their corresponding parameters among different cliques in a graph. These sets are known as *clique templates*. An example of a shared factor in a clique template would be the factors $\Psi_t(y_{t-1}, y_t, x_t)$, extensively used in linear chains for entity recognition to locate and classify named entities in text into predefined categories, such as the names of persons, organizations, locations, etc. This type of factor take into account the classification of the current (y_t) and the previous (y_{t-1}) word in a sentence. This feature function and its corresponding parameter can be clearly shared throughout any position in a text.

Making use of clique templates, the set of cliques $\mathcal{C}$ can be divided in a factor graph G into $\mathcal{C} = \{C_1, ..., C_p\}$ clique templates. All cliques in each clique template C_p share the same parameters Λ_p, with Λ as the set of all parameters in graph G, so a template C_p shares the feature functions $\{\lambda_{pk} f_{pk}(\boldsymbol{x}_C, \boldsymbol{y}_C)\}_{k=1}^{K(p)}$.

Equation (3-8) can then be re-written as

$$\boxed{p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{C \in \mathcal{C}} \prod_{\Psi_C \in C_p} \Psi_C(\boldsymbol{x}_C, \boldsymbol{y}_C; \Lambda_p)} \tag{3-12}$$

The partition function, in turn, is rearranged as

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}'} \prod_{C \in \mathcal{C}} \prod_{\Psi_C \in C_p} \Psi_C(\boldsymbol{x}_C, \boldsymbol{y}'_C; \Lambda_p) \tag{3-13}$$

In those two equation, considering $K(p)$ as the number of potential functions for template C_p and λ_{kp} the template-dependent weights of the potential functions, the clique potential is then

$$f_C(\boldsymbol{x}_C, \boldsymbol{y}_C) = \exp\left(\textstyle\sum_{k=1}^{K(p)} \lambda_{kp} f_{kp}(\boldsymbol{x}_C, \boldsymbol{y}_C)\right)$$

3.2.1 Parameter estimation

Here, we discuss the case where the training and the testing data are independent and the training data is fully observed. The $\Lambda = \{\lambda_1, ..., \lambda_{kp}\}$ parameters estimation is achieved by maximizing the conditional likelihood.

To simplify derivations, we maximize the equivalent conditional log-likelihood:

$$\begin{aligned}\mathcal{L}(\Lambda) &= \log\left(\frac{1}{Z(\boldsymbol{x})}\exp \textstyle\sum_{C\in\mathcal{C}}\sum_{\Psi_C\in C_p}\sum_{k=1}^{K(p)}\lambda_{kp}f_{kp}(\boldsymbol{x}_C,\boldsymbol{y}_C)\right)\\ &= \sum_{C\in\mathcal{C}}\sum_{\Psi_C\in C_p}\sum_{k=1}^{K(p)}\lambda_{kp}f_{kp}(\boldsymbol{x}_C,\boldsymbol{y}_C) - \log Z(\boldsymbol{x})\\ &= \sum_{C\in\mathcal{C}}\sum_{\Psi_C\in C_p}\sum_{k=1}^{K(p)}\lambda_{kp}f_{kp}(\boldsymbol{x}_C,\boldsymbol{y}_C) - \log\sum_{\boldsymbol{y}}\sum_{C\in\mathcal{C}}\sum_{\Psi_C\in C_p}\sum_{k=1}^{K(p)}\lambda_{kp}f_{kp}(\boldsymbol{x}_C,\boldsymbol{y}_C)\end{aligned}$$

The *gradient* of the conditional log-likelihood is calculated as

$$\frac{\partial(\mathcal{L}(\Lambda))}{\partial(\lambda_{kp})} = \sum_{\Psi_C\in C_p} f_{kp}(\boldsymbol{x}_C,\boldsymbol{y}_C) - \sum_{\Psi_C\in C_p}\sum_{\boldsymbol{y}'_C} f_{kp}(\boldsymbol{x}_C,\boldsymbol{y}'_C)p_\Lambda(\boldsymbol{y}'_C|\boldsymbol{x}) \tag{3-14}$$

The first term on the right side of Equation (3-14) represents the expected value of the training samples empirical distribution $p(\boldsymbol{x},\boldsymbol{y})$, while the second term corresponds to the expected value of the model distribution considering the current values of the parameters λ_k. So, the gradient measures the difference between the expected value of the features under the empirical and model distributions.

It can observed from Equation (3-14) that calculating the partition function is a hard task. If, for instance, if we have two states to be inferred, the time complexity is $O(2^M)$, considering M nodes in the graph. As a consequence, scientists usually resort to model approximations, notably pseudolikelihood [64].

To avoid overfitting, it is necessary to add a regularization parameter to Equation (3-14), which penalize parameters with too large norms. A common choice is to assume a Gaussian prior over the parameters, with zero mean and σ^2 covariance:

$$p(\Lambda) \propto \exp\left(-\tfrac{\|\Lambda\|^2}{2\sigma^2}\right) = -\textstyle\sum_{k=1}^{K(p)}\frac{\lambda_{kp}^2}{2\sigma^2}$$

The regularized conditional log-likelihood is then

$$\mathcal{L}(\Lambda) = \sum_{C\in\mathcal{C}}\sum_{\Psi_C\in C_p}\sum_{k=1}^{K(p)}\lambda_{kp}f_{kp}(\boldsymbol{x}_C,\boldsymbol{y}_C) - \log Z(\boldsymbol{x}) - \sum_{k=1}^{K(p)}\frac{\lambda_{kp}^2}{2\sigma^2} \tag{3-15}$$

Finally, training a CRF means finding the parameters Λ^* that gives the best possible prediction $\hat{y}$ [65], so these parameters can be estimated from

Equation (3-15) using *maximum-a-posteriori* (MAP) inference:

$$\Lambda^* = \underset{\Lambda}{\operatorname{argmax}}\, \mathcal{L}(\Lambda) \tag{3-16}$$

Equation (3-16) configures an optimization problem. The usual solver for parameter estimation of CRFs is Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) [66].

After estimating the parameters, it is necessary to perform the inference of each node state in the graph, described in the next section.

3.2.2 Inference

The inference algorithm is going to be invoked repeatedly, once for each time the gradient in the previous section is computed.

Belief Propagation (BP) [67, 68] is a common choice for MAP inference of the most likely states in a graphical model. When there are no loops in the graph, BP provides an exact solution, which is equivalent to dynamic programming. When the graph contains loops, BP provides an approximate, but often good, solution [69], and in this case it is known as *Loopy Belief Propagation*. In the remaining of this section, the theory behind BP is explained, with focus on pairwise Markov random fields.

Belief Propagation is an iterative process where neighboring variables pass messages to each other. The messages are of the type "I, variable $\boldsymbol{x}_i$, think that you, variable $\boldsymbol{x}_j$, belong to these states with following likelihoods...". In the case of pairwise MRFs, the joint probability is

$$P(\boldsymbol{x}_1, ..., \boldsymbol{x}_n) = \frac{1}{Z} \prod_{i=1}^{n} \Psi_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \prod_{(i,j)\in E,\ i<j} \Psi_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

The $\Psi_i(\boldsymbol{x}_i, \boldsymbol{y}_i)$ represent the unary factors and the $\Psi_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ are the pairwise factors.

Messages from node i to node j are defined as $m_{ij}(\boldsymbol{x}_j)$. These messages are similar to likelihoods, in respect that they are non-negative and do not necessarily sum to 1. A high value of $m_{ij}(\boldsymbol{x}_j)$ means that node i believes some state of j to be true. Usually, all messages are initialized to 1.

Message updates of $m_{ij}(\boldsymbol{x}_j)$ follow the schematic picture in Figure 3.5.

The message update considers all the messages arriving at i, except the

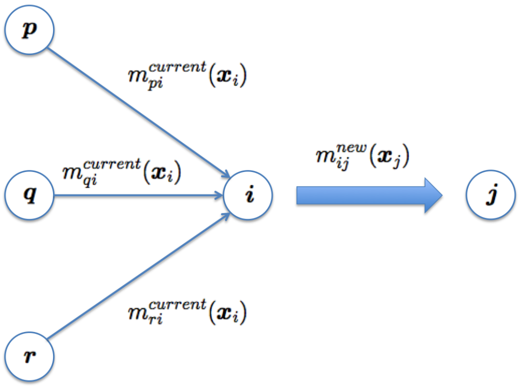

p
$m_{pi}^{current}(\boldsymbol{x}_i)$
q
$m_{qi}^{current}(\boldsymbol{x}_i)$
i
$m_{ij}^{new}(\boldsymbol{x}_j)$
j
r
$m_{ri}^{current}(\boldsymbol{x}_i)$

Figure 3.5: LBP message updating process.

message that comes from j, and so it is calculated as

$$m_{ij}^{new}(\boldsymbol{x}_j) = \sum_{\boldsymbol{x}_i} \Psi_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j)\Psi_i(\boldsymbol{x}_i) \prod_{k \in nbrs(i) \setminus j} m_{ki}^{current}(\boldsymbol{x}_i), \tag{3-17}$$

where $nbrs(i) \setminus j$ means all the neighbors of i except j.

After each iteration, it is common practice to normalize the messages to avoid overflow/underflow:

$$\sum_{\boldsymbol{x}_j} m_{ij}(\boldsymbol{x}_j) = 1$$

Once the messages have converged, the *belief state* for each node is

$$b_i(\boldsymbol{x}_i) = \Psi_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \prod_{k \in nbrs(i)} m_{ki}(\boldsymbol{x}_i) \tag{3-18}$$

The pseudo-code for belief propagation is summarized in Algorithm 1.

Algorithm 1 Belief Propagation algorithm

```
procedure BP($G(V, E)$)
    convert graph to pairwise potentials
    initialize all messages to 1
    for every $v_i$ do
        for every $v_j$ and i<j do
            repeat
                update message $m_{ij}$ according to Eq. 3-18
            until convergence
        end for
    end for
end procedure
```

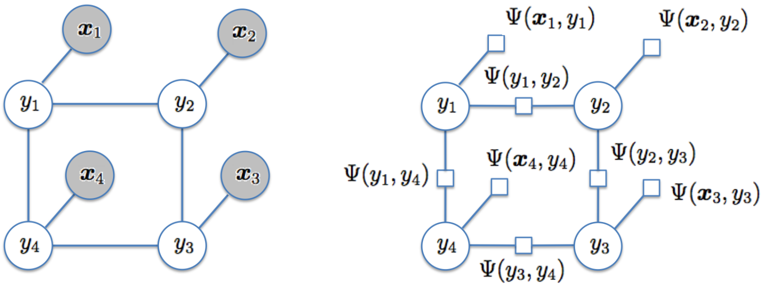
The $i < j$ statement in the algorithm is used to avoid counting each edge twice.

As an example, let's consider the simple MRF in Figure 3.6. Fig. 3.6(a) shows the Markov random field graph and Fig. 3.6(b) presents the equivalent factor graph. Variables y_i could represent segmentation labels in an image and $\boldsymbol{x}_i$ observable variables like pixel intensities.

The conditional probability function in this case is

$$P(y_1, y_2, y_3, y_4|\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) = \frac{1}{Z(\boldsymbol{x})}(\Psi(y_1, y_2)\Psi(y_2, y_3)\Psi(y_3, y_4)\Psi(y_1, y_4)$$
$$\Psi(\boldsymbol{x}_1, y_1)\Psi(\boldsymbol{x}_2, y_2)\Psi(\boldsymbol{x}_3, y_3)\Psi(\boldsymbol{x}_4, y_4))$$

At any iteration of Algorithm 1, the belief for node y_1 is obtained as



(a) MRF (b) Factor graph

Figure 3.6: A pairwise MRF and the equivalent factor graph. The gray nodes represent the observable variables.

$$b(y_1) = \Psi(\boldsymbol{x}_1, y_1)m_{41}m_{21},$$

where message from y_4 to y_1 is

$$m_{41} = \sum_{y_4} \Psi(\boldsymbol{x}_4, y_4)\Psi(y_1, y_4) \prod_{k \in nbrs(i)\backslash 1} m_{k4}(y_4)$$

and message from y_2 to y_1 is

$$m_{21} = \sum_{y_2} \Psi(\boldsymbol{x}_2, y_2)\Psi(y_1, y_2) \prod_{k \in nbrs(i)\backslash 1} m_{k2}(y_2)$$

In Chapter 4, the conditional random field concepts described here are applied in the proposed method, Transductive Conditional Random Field Regression - TCRFR.

4
Transductive Conditional Random Field Regression

In this chapter, the mathematical theory for the proposed Transductive Conditional Random Field Regression model is derived.

4.1
Basic Idea

Given a labeled sample set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^D \times \mathbb{R}\}_{i=1}^n$ and an unlabeled sample set $\mathcal{U} = \{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=n+1}^{n+m}$, let's consider a regression model with Gaussian noise:

$$y = f(\boldsymbol{x};\boldsymbol{w}) + \epsilon, \quad \epsilon = y - f(\boldsymbol{x};\boldsymbol{w}) \sim \mathcal{N}(0,\sigma^2),$$
$$p(y|\boldsymbol{x},\boldsymbol{w}) \propto \exp(-\frac{1}{2\sigma^2}|y - f(\boldsymbol{x};\boldsymbol{w})|^2) \ ,$$

where $\boldsymbol{x} \in \mathbb{R}^D$ and $y \in \mathbb{R}$ are input and output variables, respectively, and $f(\boldsymbol{x};\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$ is a linear regression function with an unknown parameter $\boldsymbol{w} \in \mathbb{R}^D$. Variable σ^2 denotes the noise variance. A Gaussian prior is assumed for $\boldsymbol{w}$:

$$p(\boldsymbol{w}) \propto \exp\left(-\frac{\lambda'}{2}\|\boldsymbol{w}\|_2^2\right) , \tag{4-1}$$

where λ is the regularization parameter.

The *maximum a posteriori (MAP)* estimator in this case is obtained by maximizing the joint distribution of $\{y_i\}_{i=1}^n$ and $\boldsymbol{w}$ (assuming *i.i.d.* data):

$$\max_{\boldsymbol{w}\in\mathbb{R}^D} p(\{y_i\}_{i=1}^n|\{\boldsymbol{x}_i\}_{i=1}^n, \boldsymbol{w})p(\boldsymbol{w}) = \prod_{i=1}^n p(y_i|\boldsymbol{x}_i, \boldsymbol{w})p(\boldsymbol{w}) \ ,$$

or, equivalently, minimizing the negative logarithm of the joint distribution $\min_{\boldsymbol{w}\in\mathbb{R}^D} \mathcal{L}_0(\boldsymbol{w})$, where

$$\mathcal{L}_0(\boldsymbol{w}) = \lambda'\|\boldsymbol{w}\|_2^2 + \sum_i \frac{|y_i - \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle|^2}{\sigma^2} \ . \tag{4-2}$$

Equation 4-2 is the standard ridge regression setting, used in many inverse problems.

While ridge regression has been proven useful in many applications, it alone would suffer from severe drawbacks that would likely deteriorate the prediction accuracy in the porosity estimation setting, namely: (a) ridge regression assumes that data is i.i.d., so no spatial connections between data points are considered and (b) the linear dependency assumption between input and output only holds for one specific regression model. The linear dependency between impedances and porosities relies upon the facies, so one ridge regression model would not be accurate for multiple facies.

Equation 4-2 is then extended threefold:

- The dependency of the regression function $f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathcal{X}$ on a latent variable $z \in \mathcal{Z}$ is explicitly modeled, using *local* joint feature maps $\Phi : \mathcal{X} \times \mathcal{Z} \to \mathcal{H}_1$ on the labeled sample set $\mathcal{S}$;
- The dependency of the inputs of the labeled and unlabeled sample sets $\mathcal{S}$ and $\mathcal{U}$ is exploited, on the basis that these samples share spatial relations that can be modeled by conditional random fields (CRF) using a *global* joint feature map $\Psi : \otimes_{i=1}^{n+m} \mathcal{X} \times \otimes_{i=1}^{n+m} \mathcal{Z} \to \mathcal{H}_2$;
- Label prediction is constricted to the unlabeled data set $\mathcal{U}$.

In other words, Φ comprises the *local features* related to each variable $\boldsymbol{x}_i$ given a latent state z_i. In contrast, Ψ comprises the *global features* that result from spatial shared relations among combinations of distinct $\boldsymbol{x}_i$ and z_i.

Note that the local Φ and the global Ψ features maps transform the original samples into reproducing kernel Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, which correspond to kernel functions [70]. This is a classical way of approaching the encoding problem for arbitrary dependencies between $\boldsymbol{x}$ and z, as it is common in the structured output literature [36].

These extensions are applied to tackle the problem of inferring latent variables under spatio-temporal structure from few precise output measurements and many noisy input measurements, a promising approach in reservoir data analysis, as explained in Section 1.1.

4.2 Proposed Method

Please refer to Table 4.1 for a summary of symbols and short descriptions used in this section.

To tackle the problem of latent dependencies in semi-supervised regression problems, Transductive Conditional Random Field Regression (TCRFR) is proposed, which consists mainly of two parts: (a) a least-squares

Symbol	Description
$\mathcal{S}$	set of labeled data $\{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^D \times \mathbb{R}\}_{i=1}^n$
$\mathcal{U}$	set of unlabeled data $\{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=n+1}^{n+m}$
$\Phi(\cdot,\cdot) \in \mathcal{H}_1$	joint feature map for ridge regression
$\Psi(\cdot,\cdot) \in \mathcal{H}_2$	joint feature map for the CRF
$u \in \mathcal{H}_1$	ridge regression parameter
$v \in \mathcal{H}_2$	CRF parameter
$\theta \in [0,1]$	trade-off between ridge regression and CRF
$\lambda \in \mathbb{R}^+$	regularization parameter for ridge regression
Γ	regularization matrix for the CRF
$K \in \mathbb{N}^+$	number of latent states $K = \vert\mathcal{Z}\vert$
$\pi_i \in \mathcal{Z}$	latent state for corresponding sample i
$\bigotimes_{i=1}^n \mathcal{X}_i$	direct product of sets $\mathcal{X}_i, \forall i = 1,\ldots,n$
$\cdot \otimes \cdot$	(flat) tensor product

Table 4.1: List of symbols used in TCRFR.

regression with parameter $\boldsymbol{u}$, conditioned on the latent states and input instances; and (b) a conditional random field with parameter $\boldsymbol{v}$ that explicitly models the dependencies of the latent variables and is conditioned on the input instances only. Both parts receive a Gaussian prior for stabilization (like in Equation (4-1)). Starting from the ridge regression likelihood in Equation (4-2), the *maximum a posteriori* estimates are given by:

$$\begin{aligned}
& \max_{\boldsymbol{u}} p(\{y_i\}_{i=1}^n|\{\boldsymbol{x}_i\}_{i=1}^n, \boldsymbol{u})p(\boldsymbol{u}) \\
\geq & \max_{\boldsymbol{u},\boldsymbol{v},\{z_i\}_{i=1}^{n+m}} p(\{y_i\}_{i=1}^n, \{z_i\}_{i=1}^{n+m}, \boldsymbol{v}|\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{u})p(\boldsymbol{u}) \\
= & \max_{\boldsymbol{u},\boldsymbol{v},\{z_i\}_{i=1}^{n+m}} p(\{y_i\}_{i=1}^n, \{z_i\}_{i=1}^{n+m}|\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{u}, \boldsymbol{v})p(\boldsymbol{u})p(\boldsymbol{v}) \\
= & \max_{\boldsymbol{u},\boldsymbol{v},\{z_i\}_{i=1}^{n+m}} p(\{y_i\}_{i=1}^n|\{z_i\}_{i=1}^n, \{\boldsymbol{x}_i\}_{i=1}^n, \boldsymbol{u})p(\boldsymbol{u}) \\
& \qquad p(\{z_i\}_{i=1}^{n+m}|\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{v})p(\boldsymbol{v}) \\
= & \max_{\boldsymbol{u},\boldsymbol{v},\{z_i\}_{i=1}^{n+m}} \prod_{i=1}^n p(y_i|z_i, \boldsymbol{x}_i, \boldsymbol{u})p(\boldsymbol{u}) \\
& \qquad p(\{z_i\}_{i=1}^{n+m}|\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{v})p(\boldsymbol{v}).
\end{aligned} \tag{4-3}$$

The probabilities are defined accordingly:

$$p(y|z, \boldsymbol{x}, \boldsymbol{u}) \propto \exp\left(-\frac{|y - \langle \boldsymbol{u}, \Phi(\boldsymbol{x}, z)\rangle|^2}{2\sigma^2}\right), \tag{4-4}$$

$$p(\boldsymbol{u}) \propto \exp\left(-\frac{\lambda'}{2}\|\boldsymbol{u}\|^2\right), \tag{4-5}$$

$$p(\{z\}_{i=1}^{n+m}|\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{v}) = \frac{\exp\left(\langle \boldsymbol{v}, \Psi(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \{z_i\}_{i=1}^{n+m})\rangle\right)}{Z(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{v})}, \tag{4-6}$$

$$p(\boldsymbol{v}) \propto \exp\left(-\frac{1}{2}\boldsymbol{v}^\top \mathit{\Gamma} \boldsymbol{v}\right), \tag{4-7}$$

where λ' and $\mathit{\Gamma} \in \mathcal{S}_{+}^{dim\mathcal{H}_2}$ (positive semi-definite matrix) are regularization constants and $Z(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{v}) = \sum_{\hat{z}\in\otimes_{i=1}^{n+m}\mathcal{Z}} \exp\left(\langle \boldsymbol{v}, \Psi(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \hat{\mathcal{Z}})\rangle\right)$ is the partition function. Variable $\hat{\mathcal{Z}}$ represents the estimated state variables. Thus, the MAP estimator for all unknown variables, including the model parameters $\boldsymbol{u} \in \mathcal{H}_1$ and $\boldsymbol{v} \in \mathcal{H}_2$, and the latent variables $\{z_i\}_{i=1}^{n+m}$ can be obtained by solving the following problem:

$$\min_{\boldsymbol{u}\in\mathcal{H}_1, \boldsymbol{v}\in\mathcal{H}_2, \{z_i\in\mathcal{Z}\}_{i=1}^{n+m}} \mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \{z_i\}_{i=1}^{n+m}), \tag{4-8}$$

where $\mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \{z_i\}_{i=1}^{n+m})$ is a convex combination of the objectives of the ridge regression model and the conditional random field. The relative weight between the regression and CRF components is given below by the θ parameter:

$$\begin{aligned}\mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \{z_i\}_{i=1}^{n+m}) = {} & \theta\mathcal{L}_{\text{rr}}(\boldsymbol{u}, \{z_i\}_{i=1}^{n}) \\ & + (1-\theta)\mathcal{L}_{\text{crf}}(\boldsymbol{v}, \{z_i\}_{i=1}^{n+m}),\end{aligned} \tag{4-9}$$

where

$$\mathcal{L}_{\text{rr}}(\boldsymbol{u}, \{z_i\}_{i=1}^{n}) = \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{1}{2}\sum_{i=1}^{n} |y_i - \langle \boldsymbol{u}, \Phi(\boldsymbol{x}_i, z_i)\rangle|^2, \tag{4-10}$$

$$\begin{aligned}\mathcal{L}_{\text{crf}}(\boldsymbol{v}, \{z_i\}_{i=1}^{n+m}) = {} & \frac{1}{2}\|\boldsymbol{v}\Gamma^{\frac{1}{2}}\|_2^2 - \langle \boldsymbol{v}, \Psi(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \{z_i\}_{i=1}^{n+m})\rangle \\ & + \log Z(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{v}).\end{aligned} \tag{4-11}$$

Here, the parameters for the regression part are re-parameterized, so that the trade-off between the regression loss and the latent structure loss is explicit.

Figure 4.1 shows the graphical representation of the TCRFR model. The conditional dependency is given by $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{z})$, where $\boldsymbol{y}$ are the labeled regression outputs (i.e., the porosities in the current problem), $\boldsymbol{x}$ are the observable input features (i.e., the impedancies) and $\boldsymbol{z}$ represent the structure connected latent states (i.e., the facies). Unlike traditional conditional random fields, TCRFR does not assume multiple pre-labeled example structures during parameter estimation. Instead, it can be viewed as ordinary ridge regression where observations $\boldsymbol{x}_i$ and corresponding regression targets y_i are coupled through latent variables z_i with few of the examples carrying ground truth label information y. Hence, there is a single structure going through all the examples contained in the training as well as the test set, which makes the model transductive by nature.

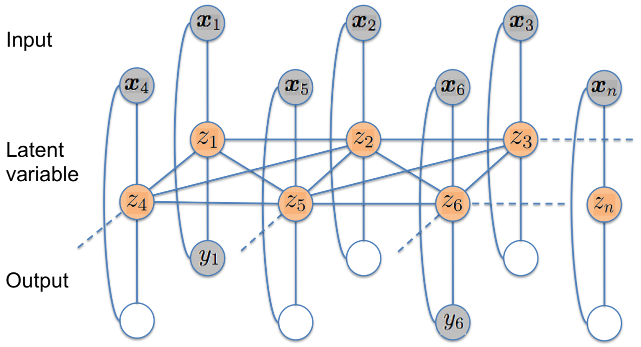


Figure 4.1: The Transductive Conditional Random Field model.

4.2.1 Optimization Scheme

To solve the non-convex problem 4-8, a Convex-Concave Procedure style scheme (CCCP) [71, 72] is adopted, which has been successfully used in structured output settings with latent variables [73]. In each (t-th) iteration, the most likely configuration $\{z_i\}$ is inferred, given $\boldsymbol{u}$ and $\boldsymbol{v}$, for all training examples,

$$\begin{aligned} \{\hat{z}_i\}_{i=1}^{n+m} &= \underset{\{z_i \in \mathcal{Z}\}_{i=1}^{n+m}}{\text{argmin}} \ \mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \{z_i\}_{i=1}^{n+m}) \\ &= \underset{\{z_i \in \mathcal{Z}\}_{i=1}^{n+m}}{\text{argmin}} \ \frac{\theta}{2} \sum_{i=1}^{n} |y_i - \langle \boldsymbol{u}, \Phi(\boldsymbol{x}_i, z_i) \rangle|^2 \\ &\quad - (1-\theta) \langle \boldsymbol{v}, \Psi(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \{z_i\}_{i=1}^{n+m}) \rangle, \end{aligned} \tag{4-12}$$

and then update the ridge regression parameter $\boldsymbol{u}$ and the CRF parameter $\boldsymbol{v}$ respectively (see Algorithm 2 for pseudo-code),

$$\hat{\boldsymbol{u}} = \underset{\boldsymbol{u} \in \mathcal{H}_1}{\text{argmin}} \, \mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \{z_i\}_{i=1}^{n}) = \underset{\boldsymbol{u} \in \mathcal{H}_1}{\text{argmin}} \, \mathcal{L}_{\text{rr}}(\boldsymbol{u}, \{z_i\}_{i=1}^{n}), \tag{4-13}$$

$$\hat{\boldsymbol{v}} = \underset{\boldsymbol{v} \in \mathcal{H}_2}{\text{argmin}} \, \mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \{z_i\}_{i=1}^{n+m}) = \underset{\boldsymbol{v} \in \mathcal{H}_2}{\text{argmin}} \, \mathcal{L}_{\text{crf}}(\boldsymbol{v}, \{z_i\}_{i=1}^{n+m}). \tag{4-14}$$

Steps (4-12) to (4-14) are summarized in Algorithm 2:

Considering the existence of a minimizer for the inference problem in Eq. (4-12), it is easy to show that, for each iteration in Algorithm 2, the objective function monotonically decreases. From the minimizer, one observes that

$$\mathcal{L}(\boldsymbol{u}^t, \boldsymbol{v}^t, \{z_i^{t+1}\}_{i=1}^{n+m}) \leq \mathcal{L}(\boldsymbol{u}^t, \boldsymbol{v}^t, \{z_i^t\}_{i=1}^{n+m}), \tag{4-15}$$

Algorithm 2 Transductive Conditional Random Field Regression (TCRFR)

```
procedure TCRFR($\mathcal{S}$,$\mathcal{U}$)
    put $t = 0$ and initialize $\boldsymbol{u}^t$ and $\boldsymbol{v}^t$ (e.g., randomly)
    repeat
        t:=t+1
        Minimize Eq. (4-8) by splitting it into 3 parts:
        (1) Update $\{z_i^t\}_{i=1}^{n+m}$ by Eq. (4-12) using the intermediate solutions $\boldsymbol{u}^{t-1}$
        and $\boldsymbol{v}^{t-1}$
        (2) Update $\boldsymbol{u}^t$ by Eq.(4-13) and $\{z_i^t\}_{i=1}^{n+m}$
        (3) Update $\boldsymbol{v}^t$ by Eq.(4-14) and $\{z_i^t\}_{i=1}^{n+m}$
    until $\forall\, i = 1, \dots, n+m :\; z_i^t = z_i^{t-1}$
    Predict unlabeled examples $\mathcal{U}$ using the inferred states $\{z_i^t\}_{i=n+1}^{m}$ and
    regression parameter $\boldsymbol{u}^t$: $y_i = \langle \boldsymbol{u}^t, \Phi(x_i, z_i^t)\rangle$
end procedure
```

and, because of the convexity of $\mathcal{L}_{\text{rr}}$ and $\mathcal{L}_{\text{crf}}$, it is also possible to verify that

$$\begin{aligned}
\mathcal{L}(\boldsymbol{u}^{t+1}, \boldsymbol{v}^{t+1}, \{z_i^{t+1}\}_{i=1}^{n+m}) &\leq \min_{\{\boldsymbol{u}\}} \mathcal{L}(\boldsymbol{u}, \boldsymbol{v}^{t+1}, \{z_i^{t+1}\}_{i=1}^{n+m}) \\
\min_{\{\boldsymbol{u}\}} \mathcal{L}(\boldsymbol{u}, \boldsymbol{v}^{t+1}, \{z_i^{t+1}\}_{i=1}^{n+m}) &\leq \min_{\{\boldsymbol{v}\}} \mathcal{L}(\boldsymbol{u}^t, \boldsymbol{v}, \{z_i^{t+1}\}_{i=1}^{n+m}) \\
\min_{\{\boldsymbol{v}\}} \mathcal{L}(\boldsymbol{u}^t, \boldsymbol{v}, \{z_i^{t+1}\}_{i=1}^{n+m}) &\leq \mathcal{L}(\boldsymbol{u}^t, \boldsymbol{v}^t, \{z_i^{t+1}\}_{i=1}^{n+m}) \\
\mathcal{L}(\boldsymbol{u}^t, \boldsymbol{v}^t, \{z_i^{t+1}\}_{i=1}^{n+m}) &\leq \mathcal{L}(\boldsymbol{u}^t, \boldsymbol{v}^t, \{z_i^{t}\}_{i=1}^{n+m})
\end{aligned}$$

So, in summary, the *log*-likelihood in Eq. (4-8) monotonically decreases for increasing number of iterations t, i.e., $\mathcal{L}(\boldsymbol{u}^{t+1}, \boldsymbol{v}^{t+1}, \{\hat{z}^{t+1}\}_{i=1}^{n+m}) \leq \mathcal{L}(\boldsymbol{u}^{t}, \boldsymbol{v}^{t}, \{\hat{z}^{t}\}_{i=1}^{n+m})$.

4.2.2 Choice of Joint Feature Maps

This section describes the joint feature maps that are used in the experiments depicted in Chapter 6 (more on joint feature maps in the Appendix, Section A.2. See also Section A.3 for indicator functions). Given an undirected graph $G = (V, E)$ with edges E and vertexes V, where each vertex corresponds to a sample and the state space is $S = \mathcal{Z}$, the global feature map Ψ is defined as

$$\Psi(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \{z_i\}_{i=1}^{n+m}) = \begin{pmatrix} \left(\sum_{(e_i,e_j)\in E} \mathbf{1}[z_{e_i} = s_i \wedge z_{e_j} = s_j]\right)_{(s_i,s_j)\in S}, \\ \left(\sum_{v\in V} \mathbf{1}[z_v = s]\, \phi(x_v)\right)_{s\in S} \end{pmatrix} \tag{4-16}$$

In this joint feature map there are two types of features, pairwise and unary.

The pairwise features are represented in the first line of the joint feature map formulation. For any edge $(e_i, e_j) \in E$, one sums up all the features that satisfy the condition that state z of node e_i (z_{e_i}) is equal to some state s_i and state z of vertex e_j (z_{e_j}) is equal to some state s_j. For each feature that satisfies the condition, its output is 1 and 0 otherwise, as defined by the indicator function $\mathbf{1}[z_{e_i} = s_i \wedge z_{e_j} = s_j]$.

The unary functions are represented in the second line of the joint feature map formulation in (4-16). For any vertex $v \in V$, one sums up all the features that satisfy the condition that state z of node v (z_v) is equal to some state s. For each feature that satisfies the condition, its output is 1 times the value of the feature $\phi(x_v)$ and 0 otherwise, as defined by the indicator function $\mathbf{1}[z_v = s]\, \phi(x_v)$.

Accordingly, the local regression joint feature map is defined Φ as

$$\Phi(\boldsymbol{x}, z) = \phi(\boldsymbol{x}) \otimes \Lambda(z), \tag{4-17}$$

where $\Lambda(z) \in \{0,1\}^K$ with entries $(\Lambda(z))_k = 1$ if $z = k$ and 0 otherwise. $K \in \mathbb{N}^+$ is the number of hidden states.

4.2.3 Latent State Inference

Latent state inference is computationally hard in general. While for tree-like structures efficient global inference schemes exist, this does not hold true for settings with loops [69]. Since the focus is on the latter, it becomes necessary to rely upon approximation methods. Two of the most used inference approximation methods are Quadratic Programming Approximation (QPA) [74] and Loopy Belief Propagation Approximation (LBPA) [75].

Because QPA is computationally demanding and does not scale well with the number of edges, in the remaining of this thesis the focus is on Loopy Belief Propagation Approximation.

4.2.3.1 Loopy Belief Propagation Approximation (LBPA)

In Loopy Belief Propagation Approximation, each $\hat{z}_i$ is sequentially updated given the states of its neighbors. This approach is proven to monotonically decrease the objective for each iteration and therefore Assumption (4-15) holds even in the presence of loops. Moreover, in case of tree-like structures, LBPA does converge to the global solution. The algorithm works by iteratively sending messages $M_{ij}(s)$ from node i to node j (in state

s) in the proximity of its location:

$$M_{ij}(s) \leftarrow \varepsilon + \max_t \iota_{ij}(s,t) + \vartheta_i(t) + \sum_{k \in N(i)/j} M_{ki}(t) \ ,$$

where ε is some normalization constant, $N(i)$ denotes the set of neighboring nodes of node i and

$$\begin{aligned} \iota_{ij}(s,t) &= (1-\theta)\boldsymbol{v}_{st}, \\ \vartheta_i(t) &= (1-\theta)\langle \boldsymbol{v}_t, \phi(\boldsymbol{x}_i)\rangle + \mathbf{1}[i \leq n]\frac{\theta}{2}|y_i - \langle \boldsymbol{u}, \Phi(\boldsymbol{x}_i, t)\rangle|^2 \ . \end{aligned}$$

After convergence, max-marginals $\mu_i(s)$ can be computed as follows,

$$\mu_i(s) \leftarrow \varepsilon + \max_t \vartheta_i(t) + \sum_{k \in N(i)} M_{ki}(t) \ .$$

Finally, backtracking using the max-marginals reveals the latent states per node. Experiments empirically showed that the quadratic approximation performs similar, but it is time-consuming, while the LBP approximation gives a reasonable performance and it is scalable.

4.2.4 Regression Parameter Estimation

The estimation (4-13) of $\boldsymbol{u}$ is simply a ridge regression problem, of which the solution is available analytically (cf. Section 4.1):

$$\frac{\partial \mathcal{L}_{rr}(\boldsymbol{u}, \{z_i\}_{i=1}^n)}{\partial \boldsymbol{u}} = 0 \Rightarrow \boldsymbol{u} = (\lambda I + \Phi\Phi^{\mathrm{T}})^{-1}\Phi\boldsymbol{y} \ ,$$

with $I \in \{0,1\}^{dim\mathcal{H}_1 \times dim\mathcal{H}_1}$ being the identity matrix, $\Phi \in \mathbb{R}^{dim\mathcal{H}_1 \times n}$ the design matrix of only the labeled samples, and $\Phi\Phi^{\mathrm{T}}$ the corresponding covariance matrix.

One fundamental assumption in the porosity estimation problem setting is the linearity of the regression model within each latent state. For this setting, the above regression model is sufficient. It is, however, relatively easy to extend to non-linear settings. For that, kernel ridge regression can be applied and solved analytically, which, nevertheless, is not in the scope of this work.

4.2.5 CRF Parameter Estimation

Problem (4-14) of $\boldsymbol{v}$ is convex and therefore a gradient-based solver with L-BFGS is used, which is the method of choice for parameter estimation of

CRFs. To perform the gradient descent, it is necessary to compute the objective function $\mathcal{L}_{\text{crf}}$ and its gradient with respect to $\boldsymbol{v}$, which is written as

$$\begin{aligned}\nabla_{\boldsymbol{v}}\mathcal{L}_{\text{crf}}(\boldsymbol{v}, \{z_i\}_{i=1}^{n+m}) &= \mathit{\Gamma}\boldsymbol{v} - \Psi(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \{z_i\}_{i=1}^{n+m}) \\ &+\mathbb{E}_{\hat{z}\sim p(\{\hat{z}_i\}_{i=1}^{n+m}|\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{v})}[\Psi(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \{\hat{z}_i\}_{i=1}^{n+m})]. \qquad (4\text{-}18)\end{aligned}$$

The objective function (4-9) contains the partition function $\log Z(\{\boldsymbol{x}\}_{i=1}^{n+m}, \boldsymbol{v})$, and the gradient (4-18) involves the expectation

$$\mathbb{E}_{\hat{z}\sim p(\{\hat{z}_i\}_{i=1}^{n+m}|\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{v})}[\Psi(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \{\hat{z}_i\}_{i=1}^{n+m})].$$

As already explained in Section 3.2.1, computation of the partition function with pairwise interaction is known to be hard. In the experiments described in Chapter 6, it was verified that approximately 85% of the time of the optimization is spent on the calculation of the partition function and less than 15% on the inference problem. Therefore, the partition function is approximated with the pseudolikelihood [64].

5
Transductive Conditional Random Field Regression Pipeline

Although the TCRFR model described in Chapter 4 is devised to handle just a small amount of labeled data, it is still not enough for realistic and extreme situations such as in the oil exploration of petroleum reservoirs, where less than 0.01% of labeled data is available from a really small number of wells.

This chapter describes a novelty process workflow for porosity estimation from seismic impedance and well porosity data, at the heart of which is the Transductive Conditional Random Field Regression model proposed in Chapter 4. The basic idea is to perform a segmentation in the 3D seismic input volume as a preprocessing step and, from this new clustered volume, determine the neighboring graph structure to be passed to the TCRFR method, which in turn estimates facies and porosity for multiple clusters. As another preprocessing step, the model is enhanced by fixing the facies in regions where the geologist feels confident about their categorical values. The *TCRFR pipeline* is shown in Figure 5.1:

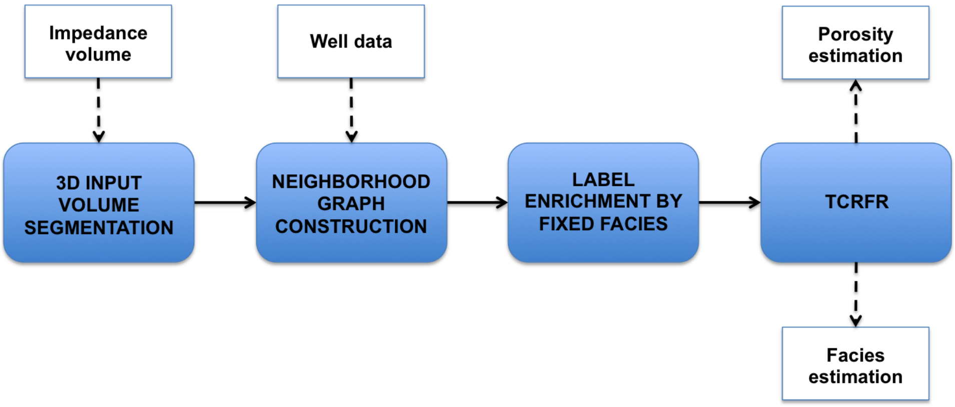

Impedance volume
Well data
Porosity estimation
3D INPUT VOLUME SEGMENTATION
NEIGHBORHOOD GRAPH CONSTRUCTION
LABEL ENRICHMENT BY FIXED FACIES
TCRFR
Facies estimation

Figure 5.1: The proposed TCRFR pipeline for porosity prediction.

Each of those steps is described in the following sections.

5.1
3D input volume segmentation

In this step, the original 2D graph-based image segmentation method proposed in [76] is extended to 3D. The goal is to define clusters (or *geobodies*, in this case) with similar features based solely on the input instances of the impedance volumes. The segmentation is applied to the whole volume. The impedance values are converted to RGB colors in a 256 color table.

The method adaptively adjusts the segmentation criterion based on the degree of variability in neighboring regions of the volume. The evidence for creating a boundary between two regions is given by comparing two quantities, one based on intensity differences across the boundary, and the other based on intensity differences between neighboring pixels within each region. Intuitively, the intensity differences across the boundary of two regions are perceptually important if they are large relative to the intensity differences inside at least one of the regions [76].

The volume is represented as a graph $G = (V, E)$, where each node $v_i \in V$ corresponds to a voxel (a value on a regular grid in 3D space) and the edges in E connect pairs of neighboring voxels $(v_i, v_j) \in E$ in a 6-connected tile, as shown in Figure 5.2(a). A weight $w(v_i, v_j)$ is associated with each edge. This weight is a non-negative measure of the dissimilarity between neighboring elements v_i and v_j based on the RGB color intensity difference between the voxels that it connects: $w(v_i; v_j) = |I(p_i) - I(p_j)|$. $I(p_i)$ represents the color intensity of voxel p_i.

The method is executed once for each of the red, green, and blue color components. Two neighboring voxels are set in the same cluster only if they independently belong to each of the same red, green, and blue clusters.

The segmentation S is a partition of V into clusters such that each cluster $C_i \in S$ corresponds to a connected component in a graph $G^{'} = (V; E^{'})$, where $E^{'} \subseteq E$. The segmentation S is not unique. The idea is that edges between two vertexes in the same component should have relatively low weights, and edges between vertexes in different component should have higher weights.

The algorithm defines a predicate D for evaluating if there is evidence to create a boundary between two clusters in a segmentation. This predicate D is based on measuring the dissimilarity between elements along the boundary of the two clusters relative to a measure of the dissimilarity among neighboring voxels within each of the two clusters. The resulting predicate compares the inter-cluster differences to the intra-cluster differences.

The *intra-cluster difference* $Int(C_i)$ of a cluster $C_i \subseteq V$ is defined as the largest weight in the minimum spanning tree MST of the corresponding graph in this cluster, so that cluster C_i only remains connected if its edge weights are less or equal to that maximum weight:

$$Int(C_i) = \max_{e \in MST(C_i, E)} w(e)$$

The *inter-cluster difference* $Dif(C_1, C_2)$ between two clusters C_1 and C_2, on the other hand, is defined as the minimum weight edge connecting these two clusters:

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j))$$

The comparison predicate D evaluates if the difference between the two components $Dif(C_1, C_2)$ is large relative to the minimum intra-difference within clusters $Int(C_1)$ and $Int(C_2)$. A threshold function τ controls the definition of "large". If the difference is large, there is evidence for a boundary between the clusters. The predicate D is then defined as

$$D(C_1, C_2) = \begin{cases} true \ \ if \ \ Dif(C_1, C_2) > MInt(C_1, C_2) \\ false \ \ otherwise \end{cases} ,$$

where the minimum internal difference $MInt$ corresponds to $MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2))$ The threshold function τ is based on the size of the component: $\tau(C_i) = k/|C_i|$, where $|C_i|$ is the size of C_i and k is a constant parameter. The parameter k corresponds to a scale factor. A larger k leads to larger components.

The algorithm implementation has three hyperparameters:

- k: the scale factor used in the threshold function;
- σ: used for smoothing the volume before the segmentation;
- min: minimum cluster size.

It is important to keep in mind that, after the segmentation step, different facies can still occur inside each cluster. This is due to the fact that the segmentation takes into account only the inputs, without considering eventual input overlaps for distinct facies in the cluster.

5.2 Neighborhood graph construction

Considering the segmentation obtained in the previous section, a neighborhood graph is created based on the clustered 3D input volume and the available wells. First, each voxel in the input volume is connected to the neighbors that belong to the same cluster in a 6-tile setting. A diagram of this 6-tile construction is shown in Figure 5.2(a). The resulting grid for a 3x3x3 volume is shown in Figure 5.2(b).

Next, additional edges are created from each labeled voxel to its surrounding neighbors that are in the same horizontal slice. These additional connections consider the neighboring voxels that belong to a circle centered on the labeled voxel, as shown in Figure 5.2(c). The white voxels correspond to the unlabeled input volume data and the blue voxels represent the labeled $(impedance, porosity)$ information from a well in the volume's resolution. As before, each edge is created only if the labeled voxel and its neighbor belong to the same cluster. The circle radius is defined as the distance from this centered labeled voxel to its closest labeled neighbor. Figure 5.2(d) shows an example of those additional connections (in red) from a labeled voxel, considering a radius equal to 2.

The additional edges help to propagate the strong information contained in the labeled samples to the surrounding neighbors. Because the areal spatial continuity is usually greater than in the vertical direction [51], only the areal (xy) information is considered to create the additional edges. The graph connection in Figure 5.2 takes into consideration the vertical and horizontal dependency entailed by Walther's law principle [77].

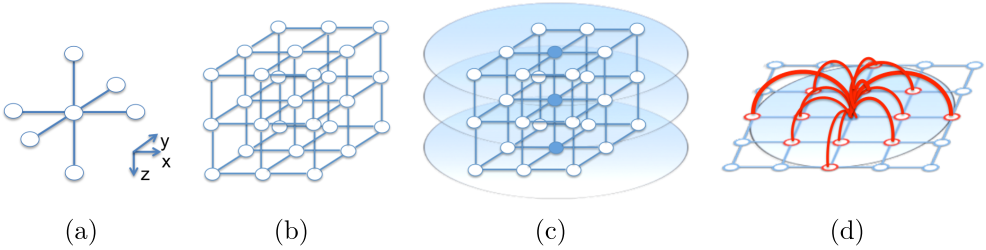


Figure 5.2: Typical 3D graph connections in a volume. Blue voxels represent the labeled samples in a well. White voxels represent unlabeled samples: (a) 6-tile voxel connections; (b) unlabeled connections in the volume; (c) radius for additional labeled (blue) to unlabeled (white) voxel connections in the horizontal slices; (d) example of connections from one labeled voxel to its neighbors, considering a radius of 2.

5.3 Labeled data enrichment by fixed facies

Since labeled data is only certain and available at the well locations, the amount of the important label information is almost negligible when compared to the bulk of unlabeled data.

Increasing the number of labeled examples greatly reduces uncertainty and leads to tighter estimates of the underlying porosity values.

Additional labels can artificially generated by:

- considering locations nearby wells as certain;
- generating various realizations for the drilled wells;

- entering manually the facies for some regions in the volume.

For the TCRFR pipeline, the focus is on the last case. The user can choose one or more time slices in the volume and then he/she roughly assigns regions in these slices which he/she is confident that belong to a specific facies. Figure 5.3 shows an example of this process. In Figure 5.3(a), the user chose one horizontal slice from the impedance volume. The red points are well locations. On top of that slice he/she drew two white stripes that represent areas which he/she is certain that belong to a non-reservoir facies. He/she also drew three black dots that correspond to a reservoir facies. Figure 5.3(b) shows the clusters resulting from the 3D input volume segmentation described in Section 5.1. Figure 5.3(c) shows the TCRFR facies estimation without considering the additional fixed facies. It can be seen that the method was not able to identify the sand channels (yellow) in some regions where there are no wells present. Figure 5.3(d) shows the TCRFR estimation considering the fixed facies provided by the user. In this case, TCRFR was able to correctly map the sand channels.

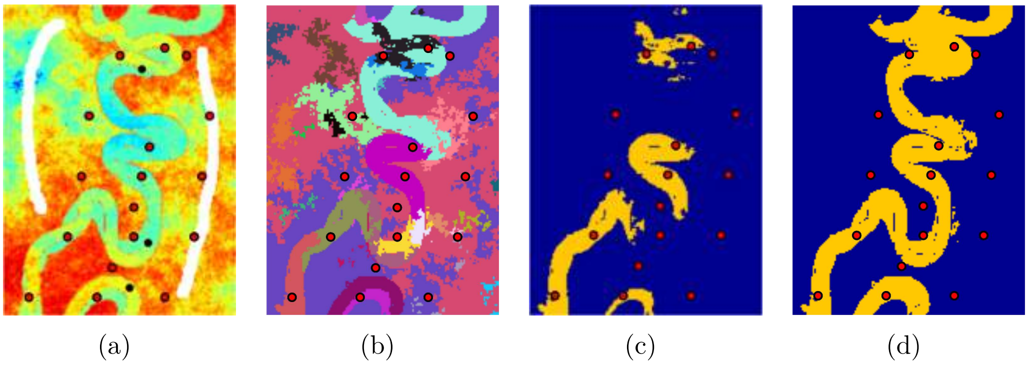


Figure 5.3: Label enrichment by fixing known facies regions specified by an expert: (a) impedance input slice with well locations (red) and fixed facies. The white stripes represent non-reservoir facies and the black dots are the reservoir facies; (b) the 3D segmentation result; (c) TCRFR facies estimation without considering the fixed facies; (d) TCRFR facies estimation considering the fixed facies.

It is interesting to notice that, because of the 3D segmentation, the information provided by the black dot positioned on the upper part of the impedance slice propagates throughout the whole cluster colored cyan, reaching the top region of the sand channel.

Geologists are used to make several assumptions about the geological model, mainly during the geologic evaluation of a reservoir in the exploration phase, where the available labeled data (i.e., porosity) is really scarce. The hand-labeling step is not necessarily a requirement for the TCRFR pipeline to work, but it can substantially improve the porosity prediction results, if the geologist detains sufficient expert knowledge to assign some hand-labeling facies. It is important to keep in mind that just one pixel in one slice in the whole volume can be already of great help for the method, as this valuable information is propagated throughout the whole 3D segment to which that pixel (or voxel, in fact) belongs to in the volume, due to the graph structure.

5.4 Transductive Regression with Latent Dependencies and multiple clusters

Here, the TCRFR method is extended twofold: (1) it is assumed that the graph can be decomposed into multiple, independent sub-graphs, and (2), some of the latent states can be fixed in advance. It is shown empirically in Chapter 6 that these extensions enable TCRFR to be applied to higher number of data points as well as fewer labeled examples without suffering from accuracy decrease.

Originally, TCRFR consists mainly of two parts: (a) a least-squares regression part with parameter $\boldsymbol{u} \in \mathcal{H}_1$, conditioned on the latent states and input instances; and (b) a conditional random field part with parameter $\boldsymbol{v} \in \mathcal{H}_2$ that explicitly models the dependencies of the latent variables and is conditioned only on the input instances. Both parts receive a Gaussian prior for stabilization and the focus is on the *maximum a posteriori* (MAP) estimates.

Building upon Chapter 4, $\mathcal{S} := \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^D \times \mathbb{R}\}_{i=1}^{n}$ labeled data points and $\mathcal{U} := \{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=n+1}^{n+m}$ unlabeled data points are given. Additionally, $\Phi(\cdot,\cdot) \in \mathcal{H}_1$ and $\Psi(\cdot,\cdot) \in \mathcal{H}_2$ are the joint feature maps into the feature spaces for the ridge regression and the CRF respectively. In this setting, $\lambda \in \mathbb{R}^+$ and $\mathit{\Gamma} \in \mathcal{S}_+^{dim \mathcal{H}_2}$ (positive semi-definite matrix) are regularization hyperparameters and $0 \leq \theta \leq 1$ is the trade-off hyperparameter between the regression and CRF parts. Further, let $Z(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{v}) = \sum_{\hat{\mathcal{Z}} \in \otimes_{i=1}^{n+m} \mathcal{Z}} \exp\left(\langle \boldsymbol{v}, \Psi(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \hat{\mathcal{Z}})\rangle\right)$ be the partition function. Thus, the MAP estimator for all unknown variables, including the model parameters $\boldsymbol{u} \in \mathcal{H}_1$ and $\boldsymbol{v} \in \mathcal{H}_2$, and the latent variables $\{z_i \in \mathcal{Z}\}_{i=1}^{n+m}$, can be obtained by solving the following problem:

$$\min_{\boldsymbol{u} \in \mathcal{H}_1, \boldsymbol{v} \in \mathcal{H}_2, \{z_i \in \mathcal{Z}\}_{i=1}^{n+m}} \mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \{z_i\}_{i=1}^{n+m}) \quad \text{subject to} \quad \{z_k\}_{k \in \mathcal{M}} = \Omega \tag{5-1}$$

where $\mathcal{M}$ is the index set of fixed latent states and $\Omega \in \mathcal{Z}^{|\mathcal{M}|}$ the corresponding set of states. Equation (4-9) is repeated here, where $\mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \{z_i\}_{i=1}^{n+m})$ is the convex combination of the objectives of the regression model and the conditional random field:

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \{z_i\}_{i=1}^{n+m}) = \theta \mathcal{L}_{\mathrm{rr}}(\boldsymbol{u}, \{z_i\}_{i=1}^{n}) + (1-\theta)\mathcal{L}_{\mathrm{crf}}(\boldsymbol{v}, \{z_i\}_{i=1}^{n+m}), \tag{5-2}$$

with

$$\begin{aligned} \mathcal{L}_{\mathrm{rr}}(\boldsymbol{u}, \{z_i\}_{i=1}^{n}) =& \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{1}{2}\sum_{i=1}^{n} |y_i - \langle \boldsymbol{u}, \Phi(\boldsymbol{x}_i, z_i)\rangle|^2, & \text{(5-3)} \\ \mathcal{L}_{\mathrm{crf}}(\boldsymbol{v}, \{z_i\}_{i=1}^{n+m}) =& \frac{1}{2}\|\boldsymbol{v}\Gamma^{\frac{1}{2}}\|_2^2 - \sum_{c=1}^{C} \langle \boldsymbol{v}, \Psi(\{\boldsymbol{x}_i\}_{i\in\mathcal{I}_c}, \{z_i\}_{i\in\mathcal{I}_c})\rangle \\ & + \log Z(\{\boldsymbol{x}_i\}_{i=1}^{n+m}, \boldsymbol{v}). & \text{(5-4)} \end{aligned}$$

Equation (5-3) is the same as 4-10 and Equation (5-4) now incorporates the results from the graph-based segmentation. Variable C denotes the number of segments, and the disjoint sets $\mathcal{I}_c$, for $c = 1, \ldots, C$ correspond to the indexes within each of the segments. The joint feature map $\Psi(\{\boldsymbol{x}_i\}_{i\in\mathcal{I}_c}, \{z_i\}_{i\in\mathcal{I}_c})$ is constructed according to the neighborhood graph construction explained in Section 5.2.

For drastically saving computation time, the MAP inference is skipped for some segments, as follows.

Each segment $c = 1, \ldots, C$ satisfies one of the following conditions:

(a) The segment contains no labeled samples nor hand-labeled voxels;

(b) The segment contains only a single labeled sample or voxel hand-annotated to a single facies category;

(c) The segment contains multiple labeled samples and/or voxels hand-annotated to multiple facies categories.

For the segments satisfying (a), the improved TCRFR cannot do much, because the voxels in the segments are completely unlabeled. For those segments, the same facies category is assigned to all voxels in each segment by majority voting based on the impedance. Also for the segments satisfying (b), the same facies category is assigned to all voxels in each segment, however, in this case, the category is the one estimated for the labeled voxel from the porosity, or the one given to the hand-annotated voxels in the segment. The full MAP-inference is applied only to the segments satisfying (c).

Algorithm 3 Transductive Conditional Random Field Regression (TCRFR) with independent partitions and partially fixed latent states

```
procedure TCRFR($\mathcal{S}$,$\mathcal{U}$)
    repeat
        t:=t+1
        $c = 0$
        repeat
            Update $\{z_i^t\}_{i \in I_c}$ according to setting (a), (b), or (c) for the current partition using the intermediate solutions $\boldsymbol{u}^{t-1}$ and $\boldsymbol{v}^{t-1}$
            $c := c + 1$
        until $c = C$
        (2) Update $\boldsymbol{u}^t$ with fixed $\{z_i^t\}_{i=1}^{n+m}$
        (3) Update $\boldsymbol{v}^t$ with fixed $\{z_i^t\}_{i=1}^{n+m}$
    until $\forall\, i = 1, \ldots, n+m :\ z_i^t = z_i^{t-1}$
    Predict unlabeled examples $\boldsymbol{x}_i = n+1, \ldots, n+m$ using the inferred states $\{z_i^t\}_{i=n+1}^{n+m}$ and regression parameter $\boldsymbol{u}^t$: $y_i = \langle \boldsymbol{u}^t, \Phi(x_i, z_i^t) \rangle$
end procedure
```

Since the inference for the cases (a) and (b) can be done with *constant* time complexity, and those cases apply to many segments under very sparsely labeled scenario, this strategy provides a huge boost in runtime performance.

For the case (c), the optimization is performed in a similar fashion to the original TCRFR. The algorithm is described in Algorithm 3.

6
Experiments

In this chapter, experiments for porosity prediction are conducted on synthetic and real-world data. These experiments are divided in two parts. In the first part, the performance of the Transductive Conditional Random Field Regression method alone is evaluated, as described in Chapter 4. In the second part, the results obtained from the application of the whole TCRFR Pipeline for real-world scenarios are presented, as described in Chapter 5.

The datasets used in the experiments are described in the next section.

6.1
Dataset Description

6.1.1
Synthetic dataset

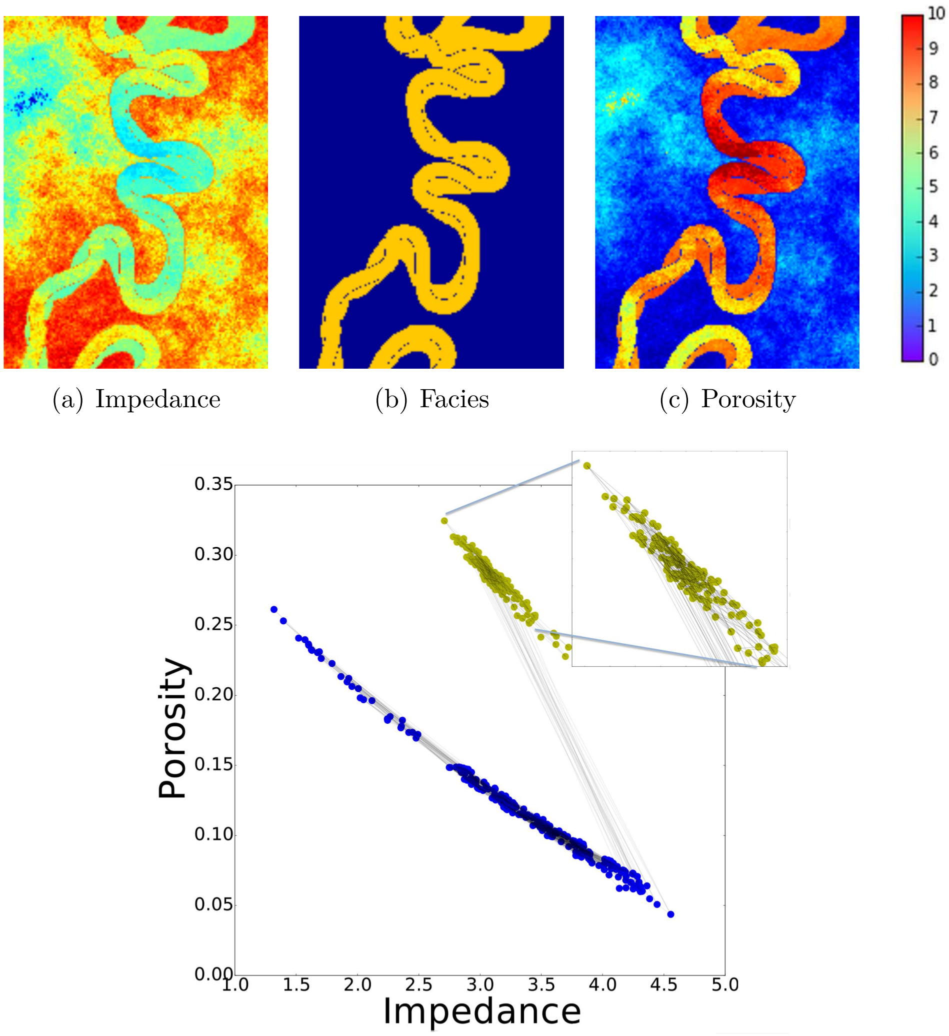
For the synthetic dataset experiments, the Stanford VI 3D reservoir benchmark dataset [4] (150 × 200 × 40 voxels) was used, based on realistic geological modeling. For reservoir exploration purposes, it is enough to segment the meandering depositional system from the shale in this example [78], so the data model was simplified for these experiments by merging the point bar and channel sands in one facies (sand), and the floodplain and boundary in another one (shale).

For easier comprehension, Figure 1.5 is repeated here. Figure 6.1 shows one horizontal data slice with 150 × 200 voxels.

In this example there are two facies, the sand channels in yellow and the background shale in blue, as seen in Fig. 6.1(b).

From the data, it is possible to observe the following trend: the sand channels have higher porosity (Fig. 6.1(c)) than the background shale, and the impedance (Fig. 6.1(a)) has a negative correlation with porosity (see also Fig. 6.1(d)).

Throughout the remaining of this chapter, the synthetic reservoir dataset is referred as Stanford-VI.



(a) Impedance

(b) Facies

(c) Porosity

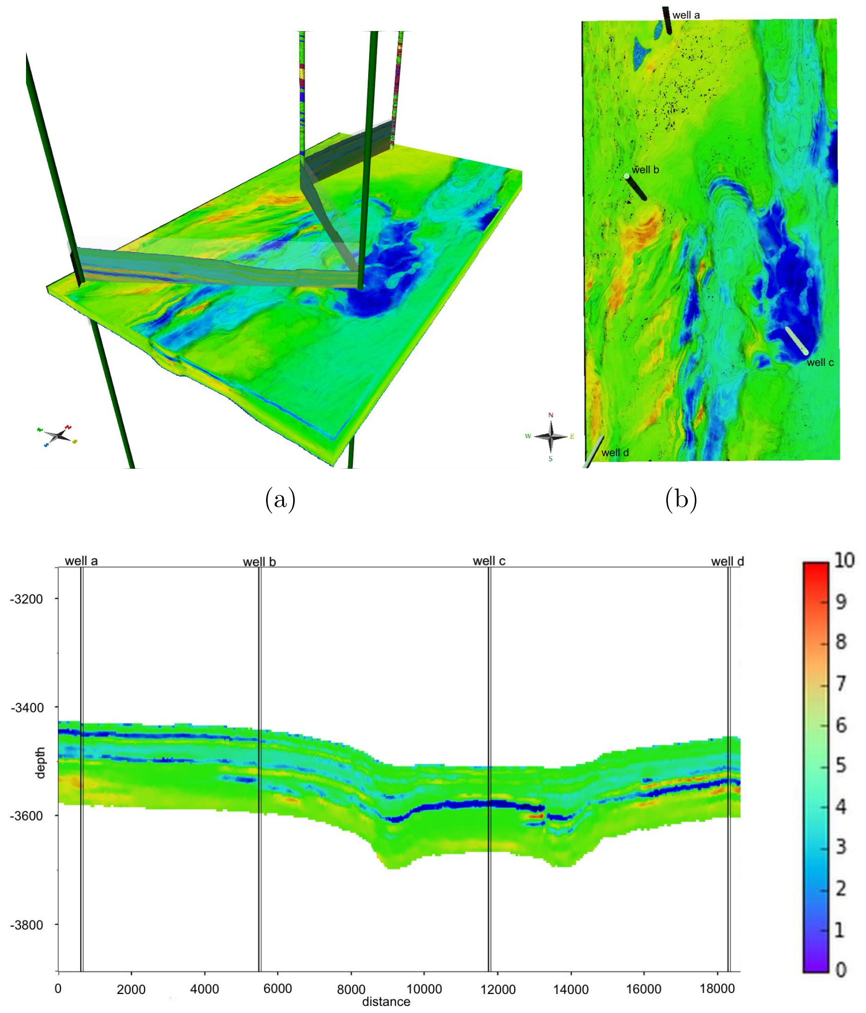
(d) input vs. output

Figure 6.1: The porosity estimation problem for one horizontal slice in the synthetic volume. In (a), the seismic input data; in (b), the related facies; (c) the corresponding porosity output; and (d) the scatter plot input × output.

6.1.2 Real-world dataset

The real-world dataset used in the experiments consists on a carbonate reservoir located in the offshore coast of Brazil (cf. Figure 6.2). It covers an area of approximately 100 square kilometers, with 460 meters in depth.

The reservoir is part of the sedimentary rock formation whose depositional model is presented in Figure 6.3. It comprises a carbonate platform with progressive shallowing cycles strongly related to subsidence, salt tectonics and sea level oscillations. The reservoir is composed of oolitic/oncolytic calcarenites developed in high energy environments (oolitic shoals). These shoals were developed in the highest parts of the structures,



(a)

(b)

(c)

Figure 6.2: The real-world data reservoir: (a) 3D view of an acoustic impedance subvolume in the reservoir with a cut section passing along the four wells; (b) map view; (c) section view passing along the four wells.

generated by the movement of the salt that accumulated in that area during middle Albian. The variations in the tectonic regime and/or fluctuations of the sea level promoted the cyclicity in the depositional system characterized by the intercalation of sediments with high and low energy. The extensive deposits of low energy, formed during high-sea level, correspond to seals to distinct reservoir units. Three facies groups occur in this region: grainstone at the bar crests with high energy sediments; oolitic/oncolytic packstones with moderate to low energy sediments at the flanks of the shoals; and peloidal packstones and wackestones in depressions located around the bars. Clay content is quite low for this carbonate environment, but the carbonate micritical matrix produces microporosity and retains irreducible water, with similar response as clay.

The volume data in the reservoir region comprises $313 \times 549 \times 74$ voxels of

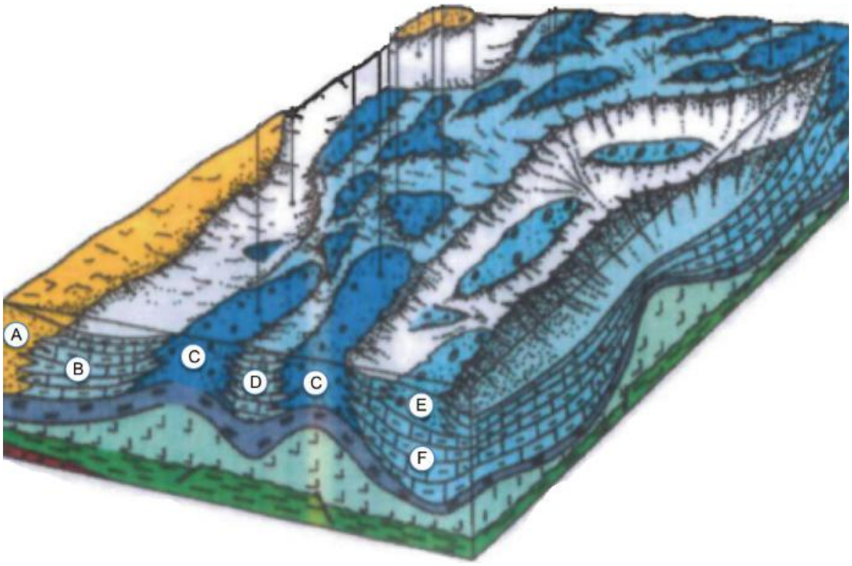


Figure 6.3: Depositional model: (A) terrigenous, tidal plain; (B) wackstones/packstones; (C) oolitic grainstones; (D) peloidal packstones; (E) oncolytic packstones; (F) wackstones/mudstones (open sea).

acoustic impedance samples. The impedance volume was previously obtained using constrained sparke-spike inversion in the Jason$^{\mathrm{TM}}$ Workbench.

Throughout the remaining of this chapter, the real-world reservoir dataset is referred as BR-1031.

6.2 TCRFR experiments

As introduced in Chapter 1, porosity estimation is a crucial step in the analysis of petroleum reservoirs for the oil industry. Although estimating porosity from seismic impedance is less accurate than from drilled wells [54], plenty of measurements are available, typically on a 3D grid covering over tens to hundreds of square kilometers.

As stated earlier, the correlation between seismic impedance and porosity depends on bodies (or units) of rock known as facies [51]. The segmentation of the reservoir into facies allows local heterogeneity and strong contrasts in rock properties to be preserved between different geological layers [79].

The proposed Transductive Condition Random Field regression model is able to simultaneously infer both facies and porosity, given as input the seismic impedances and some porosity labeled points from the available wells in the reservoir. It does not assume any prior distribution for the input data and, by definition, it can naturally handle multiple input variables/features.

In the following subsections, the performance of TCRFR and the baseline competitors are compared on the Stanford-VI and BR-1031 datasets. In all experiments, 3-fold cross validation was applied on the training samples to tune the hyper-parameters for all methods. The search range for each parameter is shown in Table 6.1. Those ranges reflect the best performance results empirically obtained for each method.

The performance is evaluated with different criteria: mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), median absolute error (MDAE), and the R^2-score. The lower bounds of the errors are also plotted for the Stanford-VI dataset, which were obtained by assuming that the latent variable was known for all the test samples.

The following baseline methods were chosen: ridge regression (RR) [80]; support vector regression (SVR) [81]; a naive approach for assessing latent states by applying k-means and using ridge regression within each cluster (k-means+RR); a mixture of experts approach (MoE) [47, 48][1]; and transductive regression (TR) [40].

Method	Parameter	Range
MoE	iterations	300, 400, ..., 800
	tolerance	1E-4, 1E-3, ..., 0.1
KMRR	ϵ	1E-5, 1E-4, 1E-3
SVR	C	1E-3, 1E-2, ..., 1.
	ϵ	0.1, 1., 10.
	kernel	linear
RR	tolerance	1E-6, 1E-5, ..., 0.1
TR	ϵ	1E-6, 1E-5, 1E-4
	C	10., 100., ..., 1E4
	C'	0.001, 0.01, ..., 1
TCRFR	R	3, 4, ..., 8
	θ	0.7, 0.75, ..., 1.0
	λ	1E-4, 1E-3, 1E-2
	γ	0.1, 1., 10.

Table 6.1: Optimized hyperparameters in the porosity prediction experiment.

As discussed in Section 1.1, no previous method has been developed for our target setting and the baseline methods above are state-of-the-art in the current research.

6.2.1 Empirical Evaluation on the Stanford-VI Dataset

In these experiments, the problem setting is simplified by only considering connections in the horizontal slices. So, from each of the given volumes,

[1] The FlexMix software package.

150×200 horizontal slices are extracted and the whole impedance data and part of the porosity data are available as the input and the regression labels (output), respectively. The goal is to infer the latent structure or, in other words, the facies, and to predict the porosities at the unlabeled samples.

The neighborhood graph is created in a 4-tile setting, connecting each pixel in a horizontal slice in the volume. The connection diagram of this 4-tile construction is shown in Figure 6.4(a). In Figure 6.4(b), the 4-tile setting is applied to a 3x3 slice of unlabeled samples. Next, additional edges are created from each labeled pixel to its surrounding neighbors in the same slice. These additional connections consider the neighboring voxels that belong to a circle centered on the labeled pixel, as shown in Figure 6.4(c). The white voxels correspond to the unlabeled input volume data and the blue voxels represent the labeled $(impedance, porosity)$ information from a well in the volume's resolution. The circle radius is defined as the distance from this centered labeled pixel to its closest labeled neighbor. Figure 6.4(d) shows an example of those additional connections (in red) from a labeled pixel, considering a radius of 2.

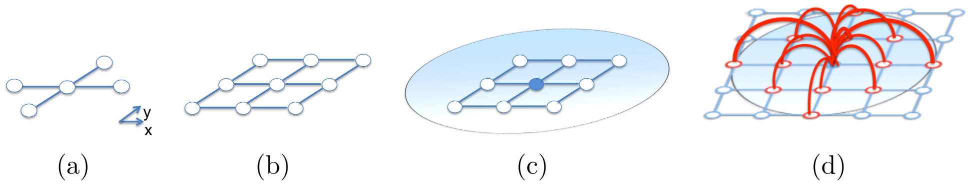


Figure 6.4: Typical 2D graph connections in a horizontal slice. Blue voxels represent the labeled samples in a well. White voxels represent unlabeled samples: (a) 4-tile voxel connections; (b) unlabeled connections in the slice; (c) radius for additional labeled (blue) to unlabeled (white) pixel connections in the slice; (d) example of connections from one labeled pixel to its neighbors, considering a radius of 2.

From the $150 \times 200 = 30,000$ pixels, 5% are randomly chosen as labeled samples and the others are treated as unlabeled samples. This process is iterated 10 times and the average performance is reported.

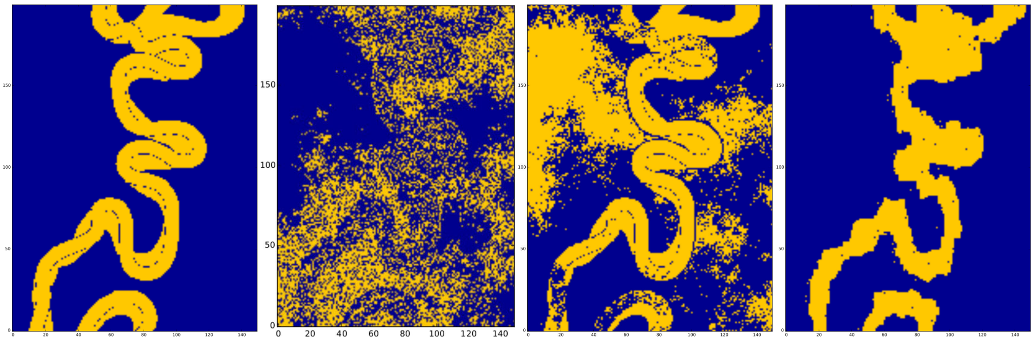
Table 6.2 summarizes the performance of TCRFR and the baseline methods. From the table it is possible to verify that TCRFR compares clearly favorably to the other state-of-the-art algorithms.

To discuss the reason of the success of TCRFR, the estimated facies and the predicted porosity for a single trial are shown in Figure 6.5 and Figure 6.6, respectively. From the competitor methods, only MoE and k-means+RR are able to provide facies estimation results.

Figure 6.5 implies that TCRFR successfully recovers the facies structure, while MoE and k-means+RR fail. The excellent facies estimation by TCRFR,

Method	MAE	MSE	RMSE	MDAE	R2
MoE	2.38477	8.44310	2.90562	1.57930	0.47237
k-means+RR	2.08030	6.27532	2.50489	1.93901	0.61407
SVR	1.84235	11.37484	3.37256	0.24478	0.28910
RR	2.05989	6.19819	2.48950	1.89004	0.61271
TR	2.05993	6.19791	2.48944	1.89106	0.61273
TCRFR	**0.69878**	**3.55215**	**1.88422**	**0.14865**	**0.77804**
L. bound	0.15237	0.03567	0.18885	0.13740	0.99777

Table 6.2: Porosity prediction performance on synthetic reservoir with 5% of labeled data for mixture of experts (MoE), Support Vector Regression (SVR), Ridge Regression (RR), Transductive Regression (TR), and Transductive Conditional Random Field Regression (TCRFR). From left to right: Mean Absolute Error (MAE); Mean Squared Error (MSE); Root Mean Squared Error (RMSE); Median Absolute Error (MDAE); Coefficient of Determination (R2).

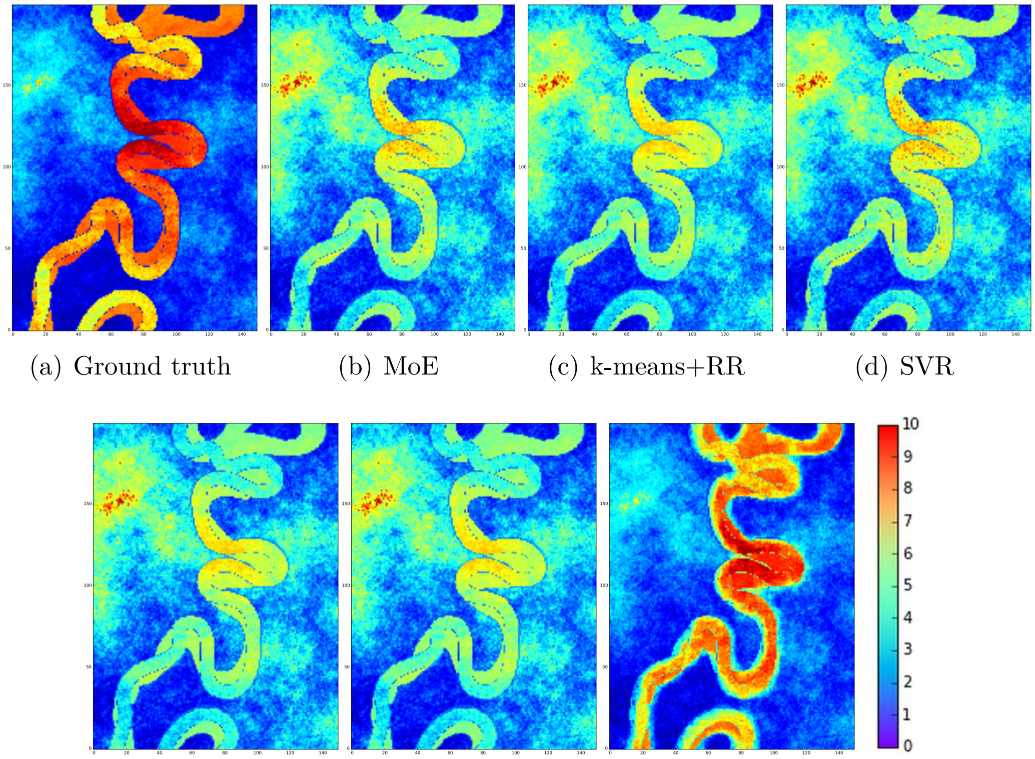


(a) Ground truth (b) MoE (c) kmeans+RR (d) TCRFR

Figure 6.5: Facies estimation results for 5% of labeled examples in one slice.

despite the small fraction of labeled data, is because it acquires the facies structure with adequate strength of correlation between neighbors, through the learning process of the conditional random field. This enables appropriate propagation of the label information, which is necessary for good facies estimation from only 5% of labeled samples. On the other hand, MoE and k-means+RR are not capable to take the structure of facies into account and, therefore, although designed with multiple regression models, one for each facies in this case, they fail to identify the facies of the unlabeled samples, because no information is propagated from the labeled samples.

Thanks to the high quality of facies estimation, TCRFR provides significantly better porosity estimation results, as shown in Figure 6.6. SVR, RR, and TR are not capable to deal with multiple regression models and, therefore, do not perform as well as the TCRFR method. As already stated, these methods do not provide facies estimation results.

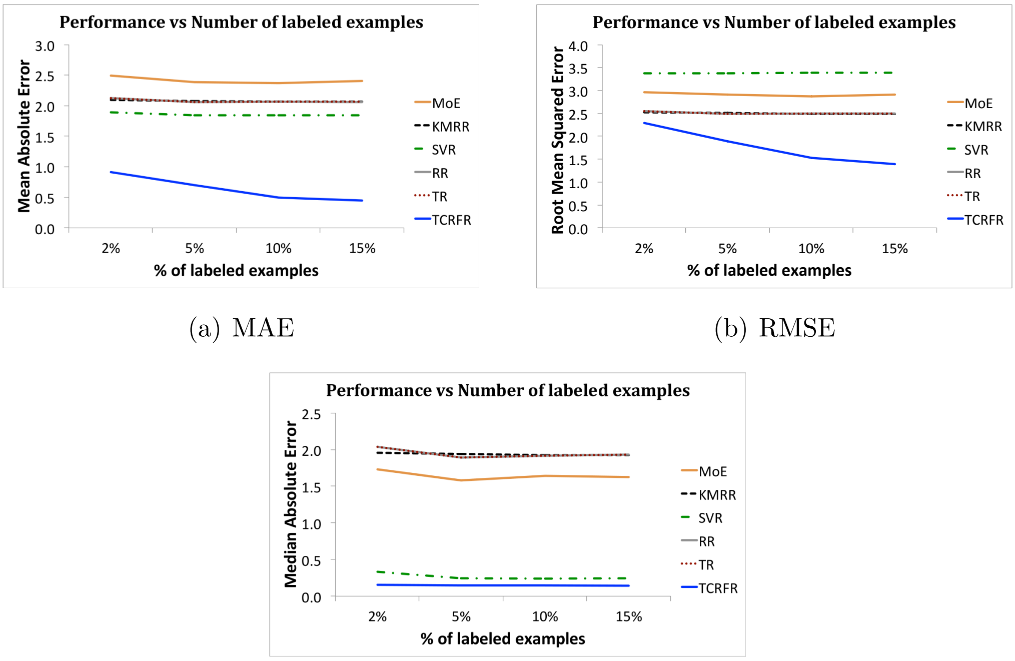


(a) Ground truth (b) MoE (c) k-means+RR (d) SVR

(e) RR (f) TR (g) TCRFR

Figure 6.6: Porosity prediction results for 5% of labeled data.

Figure 6.7 shows MAE, RMSE, and MDAE for a range of labeled samples fraction. For any fraction in this range, TCRFR outperforms all state-of-the-art competitors.

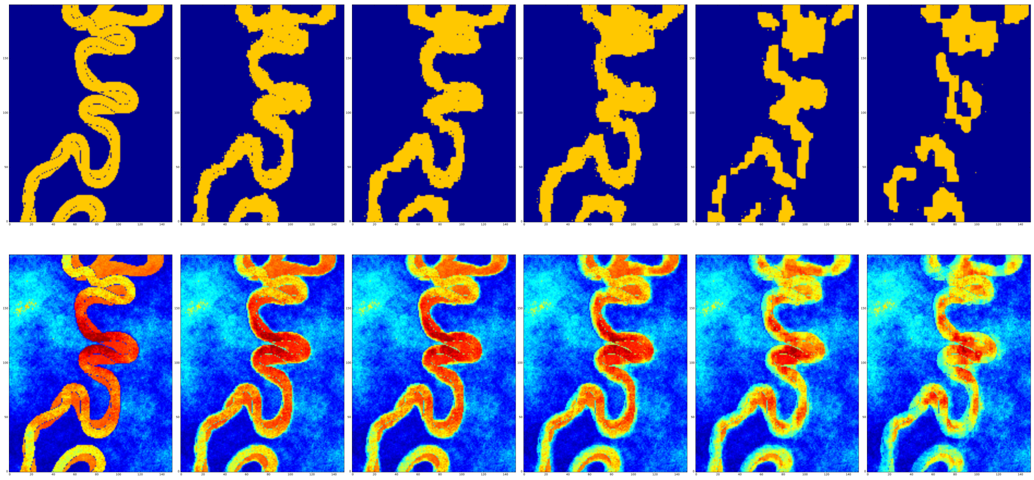


(a) MAE

(b) RMSE

(c) MDAE

Figure 6.7: MAE, RMSE, and MDAE on Stanford-VI dataset for a range of labeled data fraction.



(a) Truth (b) 15% (c) 10% (d) 5% (e) 2% (f) 1%

Figure 6.8: Estimated facies and the predicted porosity by TCRFR for different fractions of labeled samples.

Last, Figure 6.8 shows the facies estimation results (top) and the porosity prediction results (bottom) by TCRFR for different fractions of labeled samples. Although degradation is observed to some extent, TCRFR still provides reasonable facies estimation and porosity prediction, even if only

$1 \sim 2\%$ of labeled samples are available. In fact, $1 \sim 2\%$ is still much for a real-world porosity prediction application setting — only an extremely small number of labeled samples available at the drilled wells should be assumed. Section 6.3 presents the results from TCRFR Pipeline described in Chapter 5 to cope with realistic conditions and scarce labeled examples.

6.2.2 Porosity Prediction on the BR-1031 Dataset

For this real problem setting, *truly* labeled data from only four wells were available, with which no *general-purpose* machine learning method can cope. To circumvent this problem for now, additional labeled samples were created, which were obtained using geostatistical modeling (see Chapter 2 for details), specifically 3D Kriging with Locally Varying Mean (LVM).

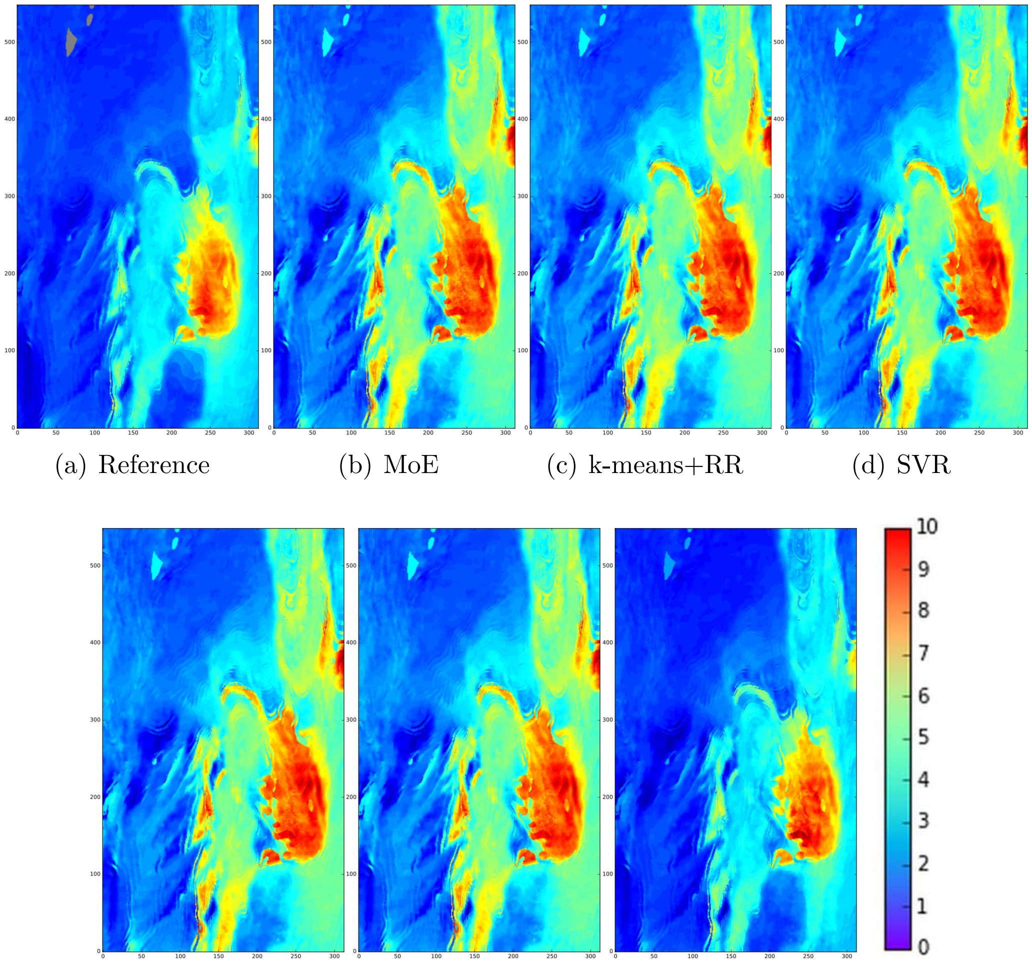
Table 6.3 shows the performance of TCRFR and the baseline methods on the BR-1031 dataset for 5% of labeled samples (including the geostatistics generated labels).

Method	MAE	MSE	RMSE	MDAE	R2
MoE	0.42502	0.55195	0.74268	0.22591	0.88991
k-means+RR	0.45002	0.44259	0.66513	0.28474	0.90910
SVR	0.48028	0.46350	0.68055	0.35463	0.90757
RR	0.45716	0.45581	0.67490	0.28999	0.90909
TR	0.45717	0.45581	0.67490	0.29000	0.90909
TCRFR	**0.24225**	**0.13712**	**0.37001**	**0.14571**	**0.97264**

Table 6.3: Porosity prediction performance on real reservoir with 5% of labeled examples.

Similarly to the experiment on the Stanford-VI dataset in the previous subsection, TCRFR compares highly favorably with the baselines.

Fig. 6.9 shows the predicted porosity by TCRFR and the baseline methods. Note that the reference in Fig. 6.9(a) is composed of the *true* labels available at the wells plus *additional* labels predicted from the geostatistics model. Again, TCRFR provides excellent results for a useful assessment of geologically attractive regions for oil exploration (red and yellow regions). Figure 6.10 shows the estimated porosity histograms and a cross plot comparing the geostatistics and TCRFR estimation results. The plots show that the results from TCRFR are similar to the ones obtained with the geostatistical approach.



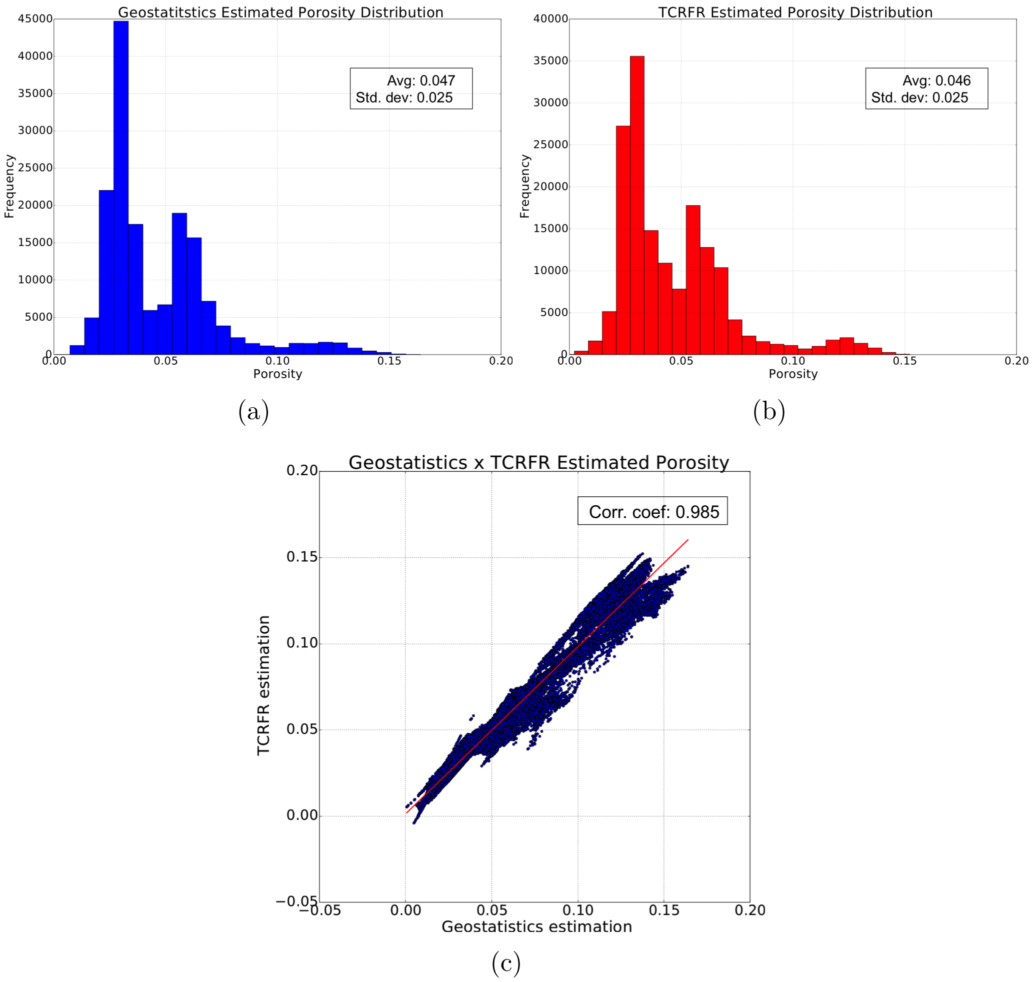
(a) Reference (b) MoE (c) k-means+RR (d) SVR

(e) RR (f) TR (g) TCRFR

Figure 6.9: Predicted porosity on the BR-1031 dataset. As a reference, the geostatistical model is used in (a).

6.3 TCRFR Pipeline experiments

In this section, experiments are again conducted on the Stanford-VI and BR-1031 reservoir datasets, but now for the whole TCRFR Pipeline. On the Stanford-VI dataset, the performance results are compared with the provided ground truth for different criteria: for prediction, the median absolute error (MDAE) and the R^2-score are used; for clustering (latent variable estimation) accuracy, the adjusted rand index (ARI) is used (more on ARI in the Appendix, Section A.4). On the BR-1031 dataset, the results obtained with the TCRFR Pipeline are compared with the ones provided by the classical geostatistics approach [13, 51]. The algorithm used was 3D Kriging with Locally Varying Mean (LVM).



(a)

(b)

(c)

Figure 6.10: Porosity statistics: (a) geostatistics porosity distribution; (b) TCRFR porosity distribution; (c) geostatistics vs. TCRFR porosity cross plot.

6.3.1 Empirical Evaluation on the Stanford-VI Dataset

The volume layer of the Stanford-VI dataset was segmented in four vertical zones (z direction) that presented distinct geometry shapes (Figure 6.11) and the TCRFFR Pipeline was then applied on each of them separately. For the input data, the shear impedance volume was used. The first row of Fig. 6.11 shows one horizontal slice with its impedance input for each of the four distinct zones. The second row in this figure presents the corresponding clustering results for each zone. The third row shows the impedance input for this slice with the annotated facies defined by the geologist. Here, the white stripes correspond to the shale facies and the black dots to the sand facies.

Porosity observation was given at 17 production wells available in the reservoir (named P1 to P6 and P21 to P31 in [4]). For each of the four zones described above, all the porosity observations at the 17 wells were used. These

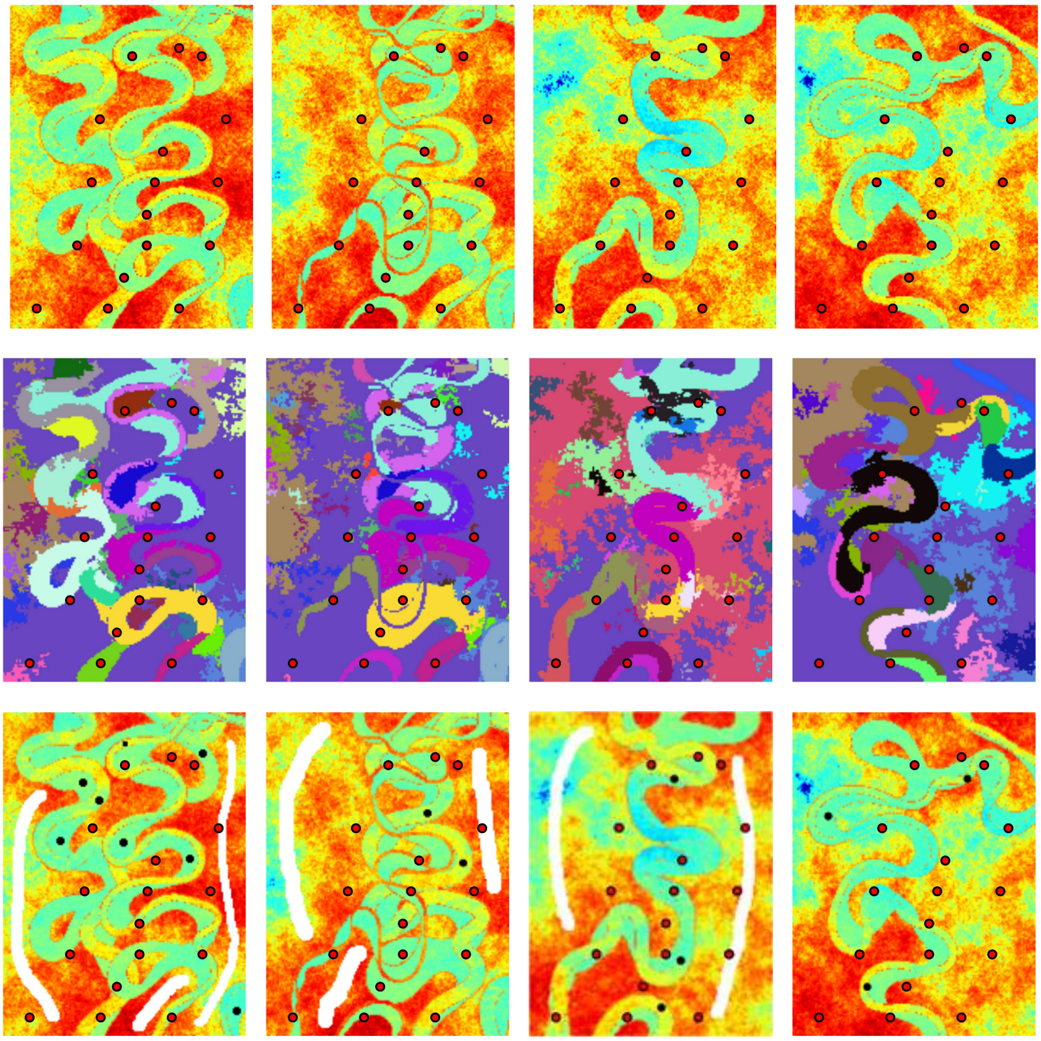


Figure 6.11: Four distinct regions used for the Reservoir-VI data experiment. The red dots correspond to the 17 well locations. Top row: the shear impedance input; Middle row: (a slice) of the graph-based segmented volume; Bottom row: manual annotations given by a geologist, where the white stripes voxels are annotated as the shale facies, and the black dot voxels are annotated as sand facies.

17 wells are represented in Figure 6.11 as red circles.

For the choice of θ and λ, leave-one-out cross-validation was applied for each of the 17 wells, obtaining $\theta = 0.99$ and $\lambda = 1$.

Figures 6.12 and 6.13 show the quality of facies and porosity estimation, respectively, by the TCRFR Pipeline. It can be seen that the TCRFR Pipeline (bottom row in each figure) accurately estimates the ground-truth (top row). Table 6.4 shows quantitative results with the performance criteria.

Zone	# Slices	MDAE	R2	ARI
1	12	0.17122	0.82054	0.68739
2	2	0.17492	0.80693	0.69699
3	10	0.16267	0.90446	0.86022
4	6	0.16527	0.91091	0.86165

Table 6.4: Performance on synthetic seismic data.

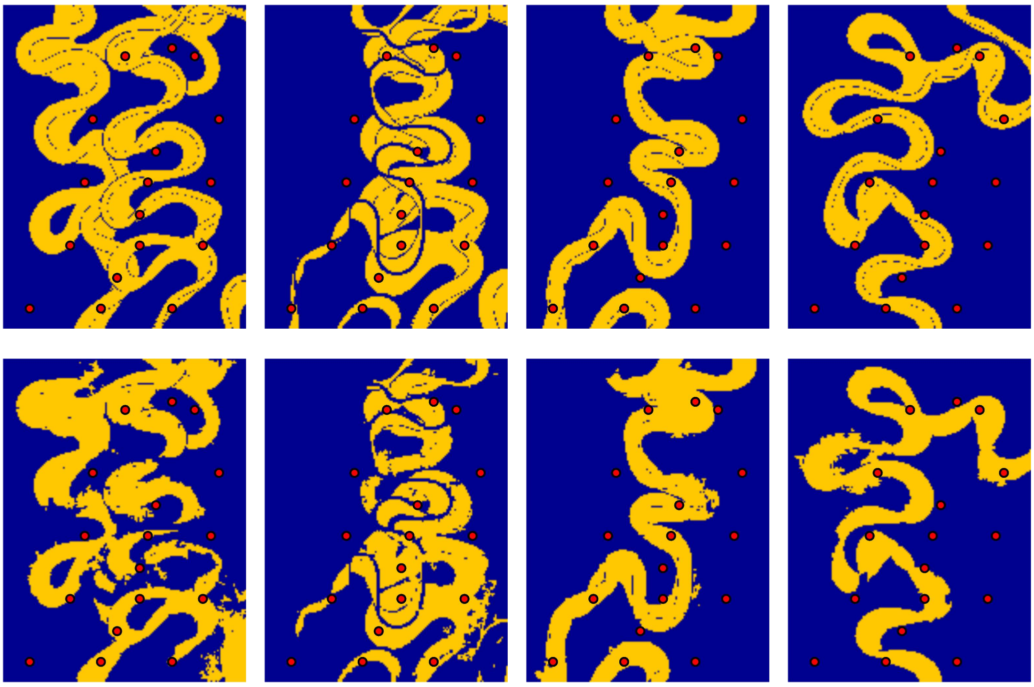


Figure 6.12: Estimated facies by the TCRFR Pipeline for 4 different regions in the Stanford-VI dataset. The red dots correspond to the well locations. Top row: the ground truth facies; Bottom row: estimated facies by the TCRFR Pipeline.

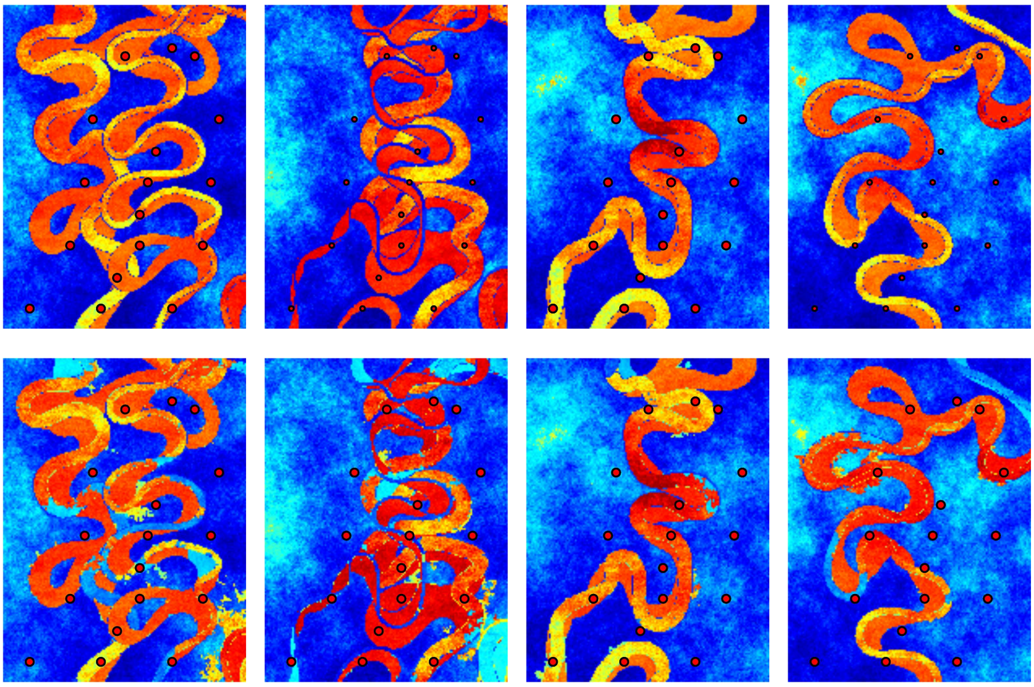
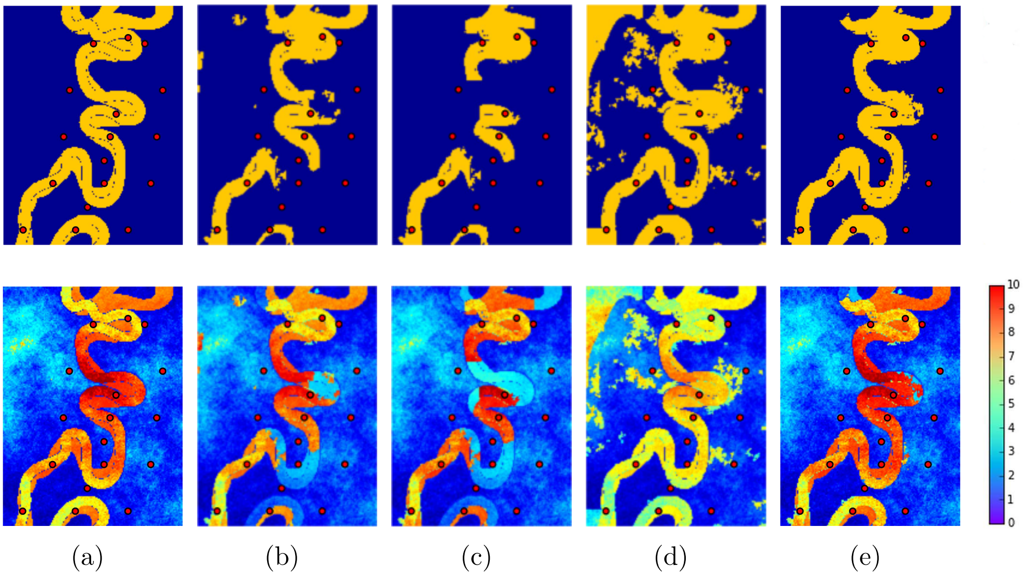


Figure 6.13: Estimated porosity by TCRFR Pipeline for 4 different regions in the Stanford-VI dataset. The red dots correspond to the well locations. Top row: the ground truth porosity; Bottom row: estimated porosity by TCRFR Pipeline.

Figure 6.14 compares the performance of the TCRFR Pipeline and the original TCRFR.[2] Comparing with the ground truth (Fig.6.14(a): facies (top) and porosity (bottom)), it can be seen that the TCRFR Pipeline (Fig.6.14(e)) outperforms the original TCRFR (Fig.6.14(b)). Figures 6.14(c) and (d) show the results with other variants of TCRFR, where just one of the new techniques, i.e., the new graph-construction and the incorporation of manual annotation, is applied. In this case, both techniques are essential for good performance of the proposed method.

In particular, in Fig. 6.14(c) the sand (yellow) channel is disconnected because of the lack of label information, which is compensated by hand-labeling in Fig. 6.14(e), while in Fig. 6.14(d) the facies of the main sand channel is accurately estimated, but the method incorrectly classifies shale (blue) regions (mainly on the top left corner of the slice) as sand. As a result, the regression model for the sand facies is inaccurately trained, which results in a poor porosity prediction over the sand channel regions. More specifically, the regression model for the sand facies is adversely affected by taking the erroneous shale high impedance values into account and, consequently, bringing

[2] Comparison with the previous methods other than TCRFR was omitted, since they were shown previously to be outperformed by the original TCRFR.

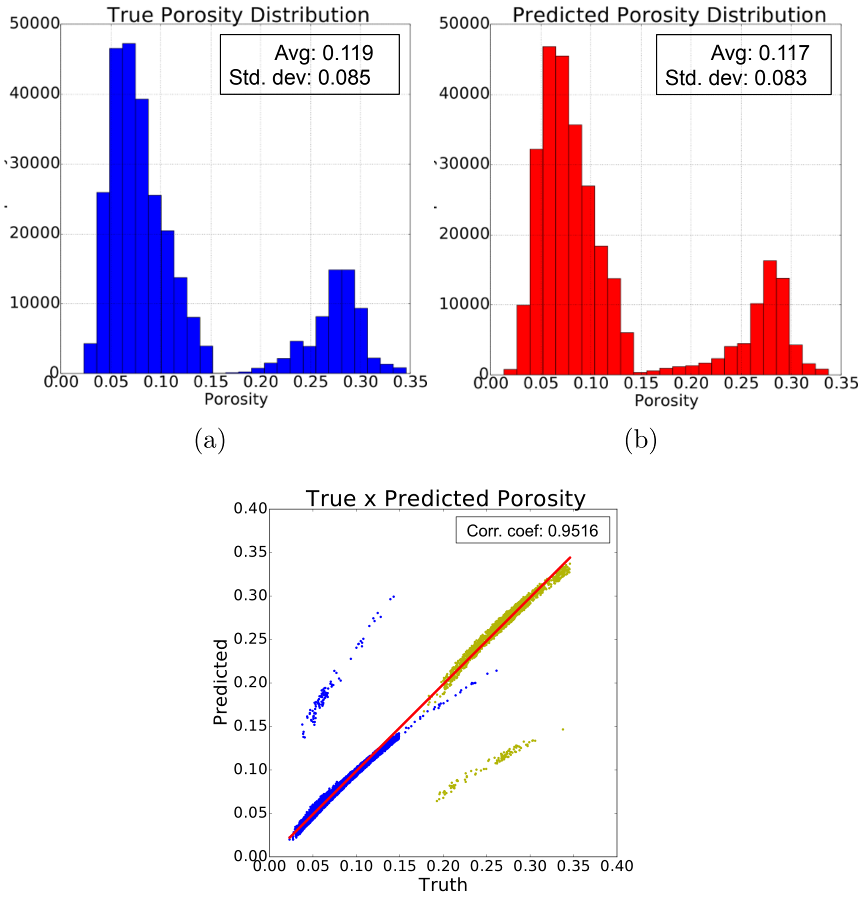


(a) (b) (c) (d) (e)

Figure 6.14: Facies (top) and porosities (bottom) results for different TCRFR methods. (a) ground truth; (b) original TCRFR; (c) TCRFR with graph construction based on the segmented volume; (d) TCRFR with manual annotations; (e) (full) TCRFR Pipeline.

the porosity down. The result is an average porosity estimation in the sand channel, which corresponds to the middle of the color table (green color).

Figure 6.15 shows some statistics of the true and the estimated porosity distributions by TCRFR Pipeline. Figs. 6.15(a) and (b) show that the distribution of the estimated porosity is quite similar to the distribution of the true one.



(a)

(b)

(c)

Figure 6.15: Porosity statistics: (a) true porosity distribution; (b) estimated porosity distribution by TCRFR Pipeline; (c) true vs. estimated porosity cross plot. Sand facies samples in yellow and shale facies samples in blue.

Fig. 6.15(c) shows that there are two small clusters with misclassified samples: the blue one on the top shows shale (low) porosity samples that were incorrectly classified into the sand facies; the yellow one on the bottom shows sand (high) porosity samples incorrectly classified in the shale facies.

TCRFR Pipeline execution time is approximately linear in the number of samples, as shown in Figure 6.16, where the method was executed varying the number of contiguous slices in the volume from one (30,000 samples) to 10 (300,000 samples).

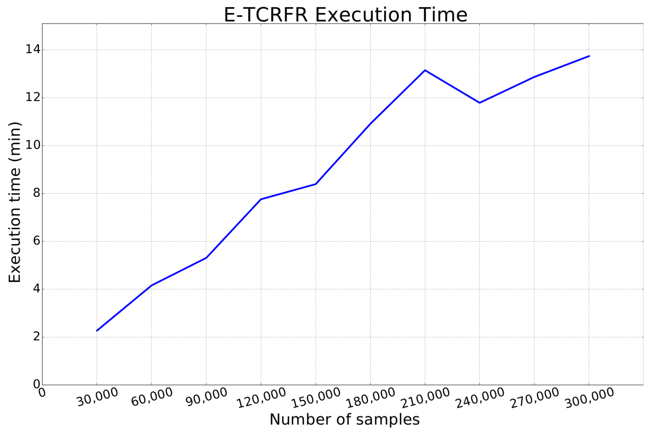
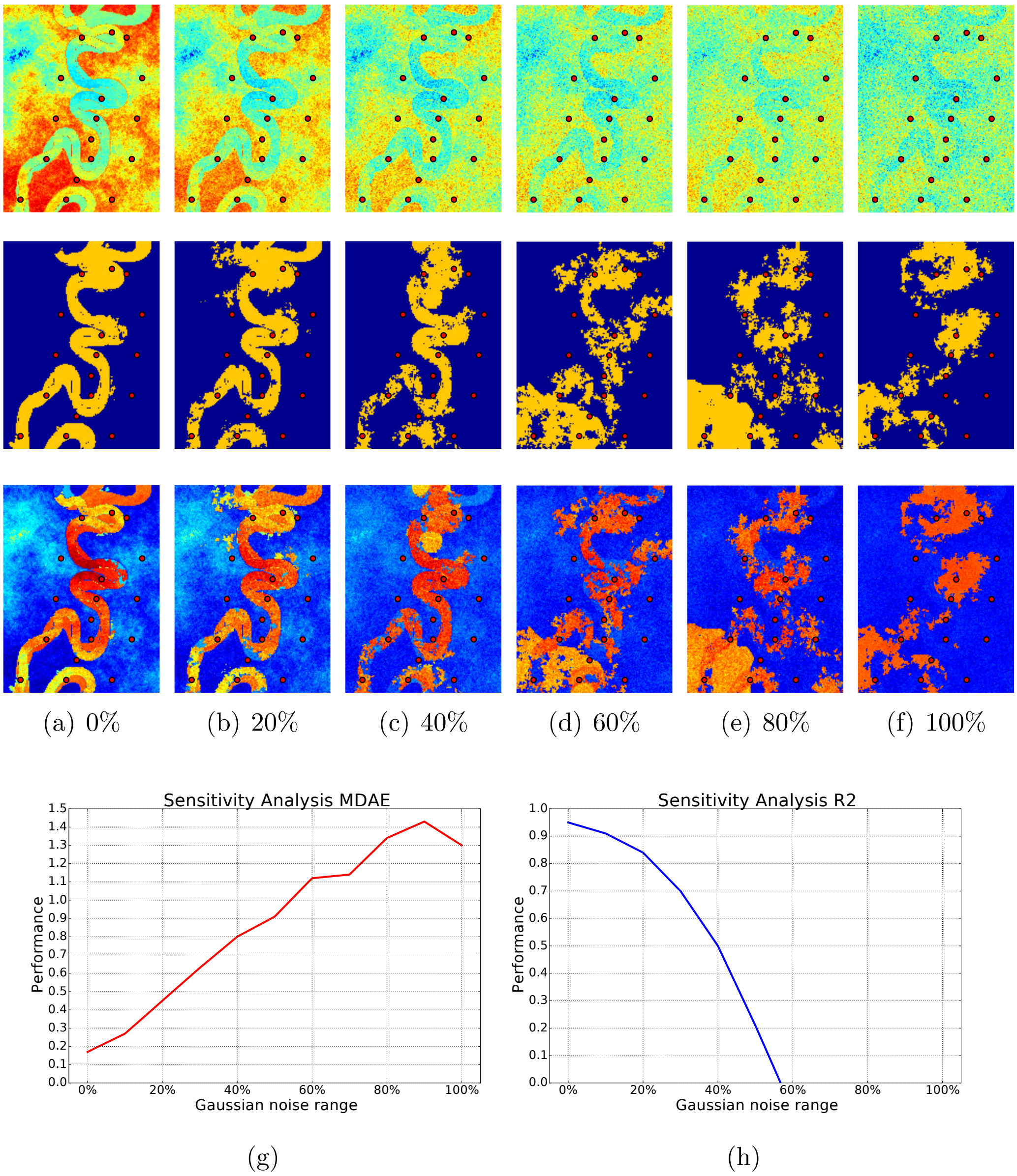


Figure 6.16: TCRFR Pipeline execution time (in minutes) from one to ten contiguous slices. Each slice contains 30,000 samples.

Sensitivity analysis was performed on a slice of the Stanford-VI dataset, progressively adding Gaussian noise over the impedance input with impedance values varying from 0 to 100%. Figure 6.17 presents the results. The top row of Fig. 6.17 (a) to (e) shows the impedance input data. The second row presents the corresponding estimated facies and the third row shows the estimated porosity. The coefficient of determination (R2) and median absolute error (MDAE) results are presented in Fig. 6.17(f). It can be observed that even with 20% Gaussian noise the R2 performance is still close to 85%, while the MDAE increases linearly with the noise.

Figure 6.18 presents another sensitivity analysis, now considering the hand-labeled facies. In Fig. 6.18(a) some portions of the sand channel are not correctly identified by the method. In Fig. 6.18(b) a black ("sand") point was added on the upper half of the slice and, as a result, a good portion of the channel is now detected by TCRFR Pipeline. Adding a second black point to the bottom half of the slice in Fig. 6.18(c) makes it possible for the method to connect the whole sand channel. Figures 6.18(d) and 6.18(e) illustrate that adding more black points to the slice do not necessarily further improve the overall result, showing that TCRFR Pipeline just requires a minimum number of hand-labeled points to provide a good performance. The MDAE and R2 plots shown in Figures 6.18(f) and 6.18(g) present the corresponding increase in the method's performance as a result of the added hand-labeled facies. Each black point in this example corresponds, in fact, to 21 pixels (facies).



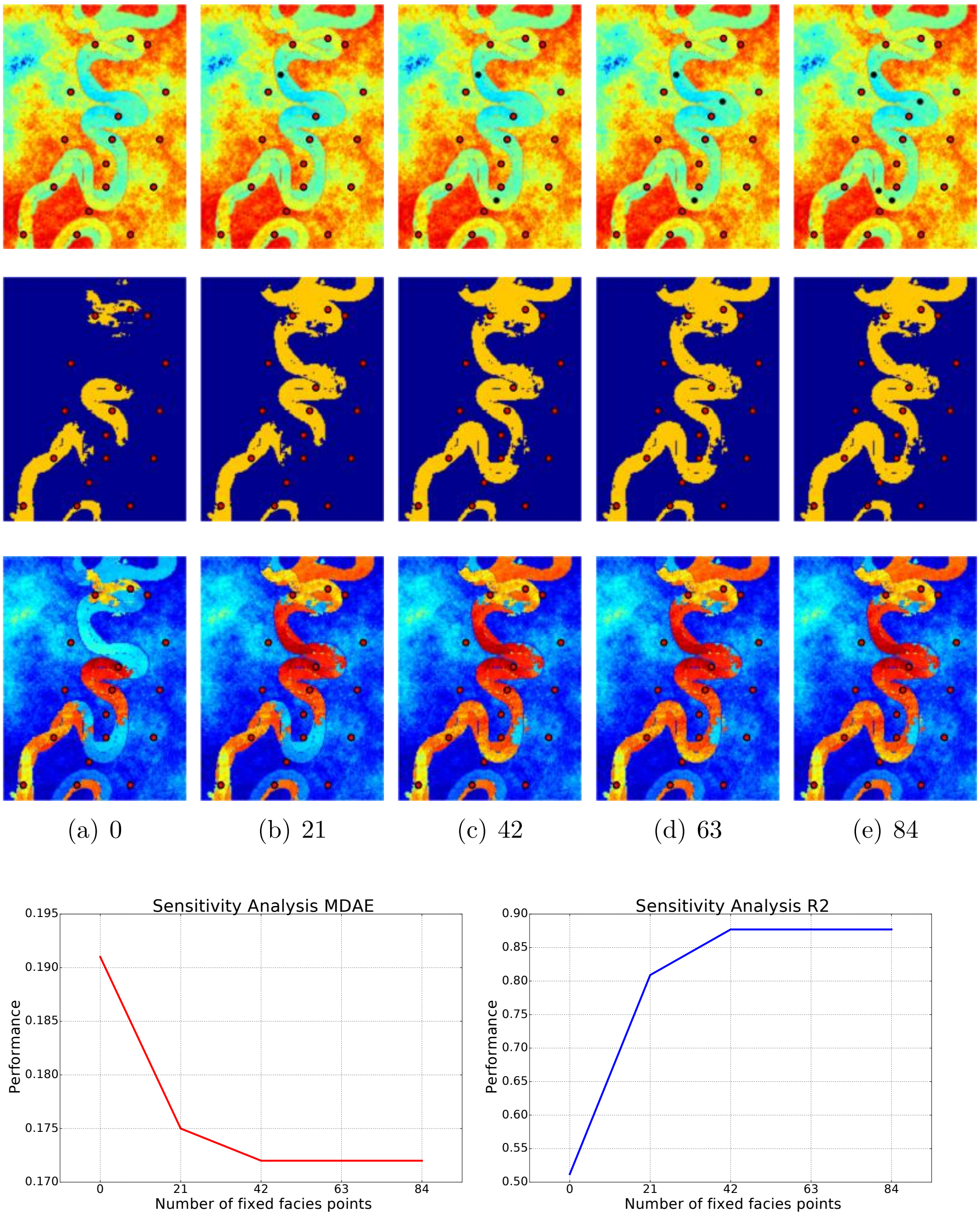
(a) 0% (b) 20% (c) 40% (d) 60% (e) 80% (f) 100%

(g) (h)

Figure 6.17: Sensitivity analysis on a slice of the Stanford-VI dataset: increasing gaussian noise applied to the impedance input, from 0 to 100% standard deviation over the original values. Top row: the impedance input; Second row: the estimated facies; Third row: the estimated porosity; Bottom row: sensitivity analysis plots for median absolute error (MDAE) and coefficient of determination (R2) with increasing gaussian noise over the original input impedance.

6.3.2
Porosity Prediction on the BR-1031 Dataset

The TCRFR Pipeline is now applied to the real carbonate reservoir already described in Section 6.1.2. From the original volume with 313 × 549 × 74 voxels of acoustic impedance samples, a subvolume was chosen with



(a) 0 (b) 21 (c) 42 (d) 63 (e) 84

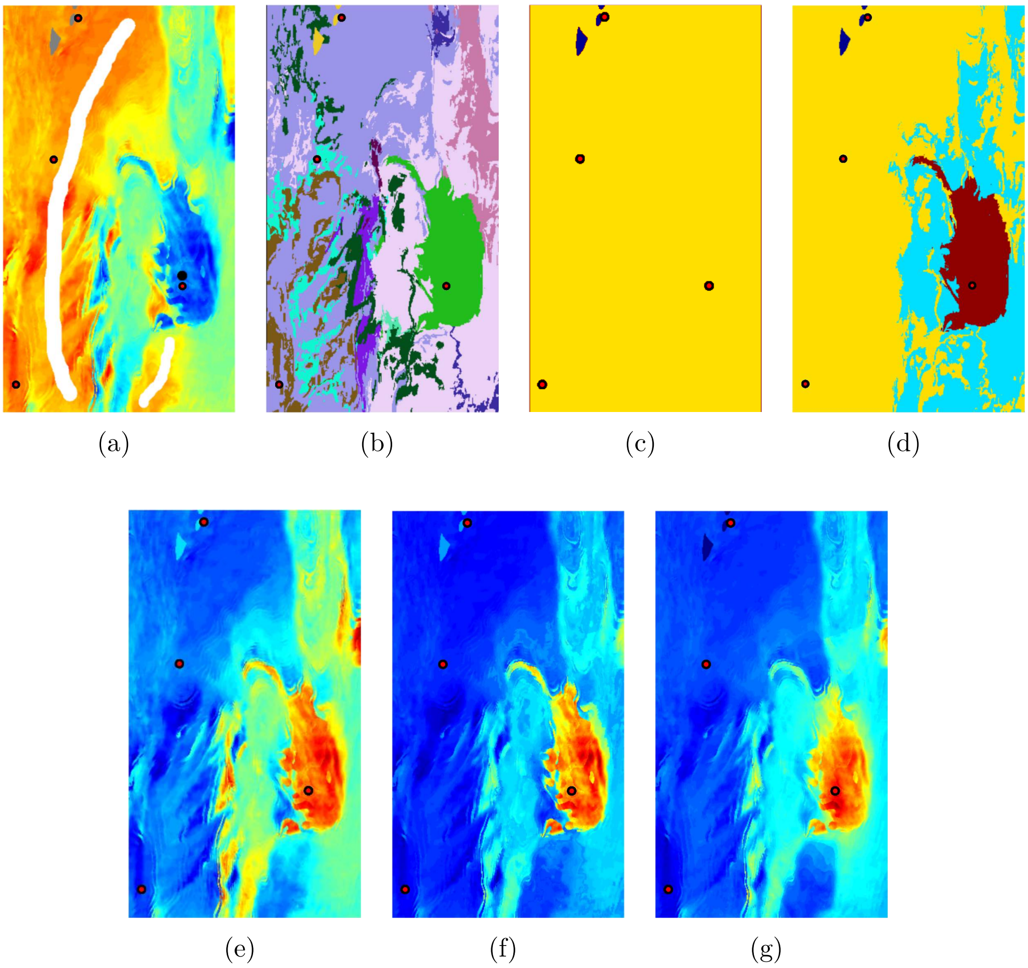
(f) (g)

Figure 6.18: TCRFR Pipeline sensitivity analysis on a slice of the Stanford-VI dataset for increasing number of hand-labeled facies defined by the geologist. Top row: the impedance input; Second row: the estimated facies; Third row: the estimated porosity; Bottom row: sensitivity analysis plots for median absolute error (MDAE) and coefficient of determination (R2) with increasing number of hand-labeled facies.

6 contiguous horizontal slices. Four exploratory wells were available with a total of 121 $(impedance, porosity)$ pair samples in seismic resolution. All the porosity values provided by the four wells were used, so the number of labeled samples correspond to approximately 0.01% of the total number of samples

in that subvolume. For comparison, porosity was also estimated with the traditional geostatistics approach. The algorithm used was 3D Kriging with Locally Varying Mean (LVM).

Figure 6.19 shows (a) a time slice of the seismic impedance input with manual annotation; (b) the graph-based segmentation result; (c) the estimated facies by the original TCRFR; (d) the estimated facies by the improved TCRFR, (e) the estimated porosity by the original TCRFR; (f) the estimated porosity by TCRFR Pipeline; and (g) the estimated porosity by geostatistics. The same hyperparameters for the Stanford-VI case were used in this experiment.



(a) (b) (c) (d)

(e) (f) (g)

Figure 6.19: Estimated facies and predicted porosity for one slice in the BR-1031 dataset: (a) impedance input with manual annotations; (b) graph-based segmentation; (c) Facies estimated by original TCRFR; (d) Facies estimated by TCRFR Pipeline; (e) Porosity estimated by original TCRFR; (f) Porosity estimated by TCRFR Pipeline; (g) porosity estimated by geostatistics.

It is possible to observe a significant gain from the original TCRFR—the improved TCRFR Pipeline (f) gives a more similar result to the geostatistics estimation (g) than the original TCRFR (e). As seen in Fig. 6.19(c), the original TCRFR is not able to correctly estimate the facies, as the number of labeled samples (121 in this case) is much smaller than the unlabeled samples (more than a million). Only one facies was found, leading to just one regression model. With the TCRFR Pipeline, three facies were estimated.

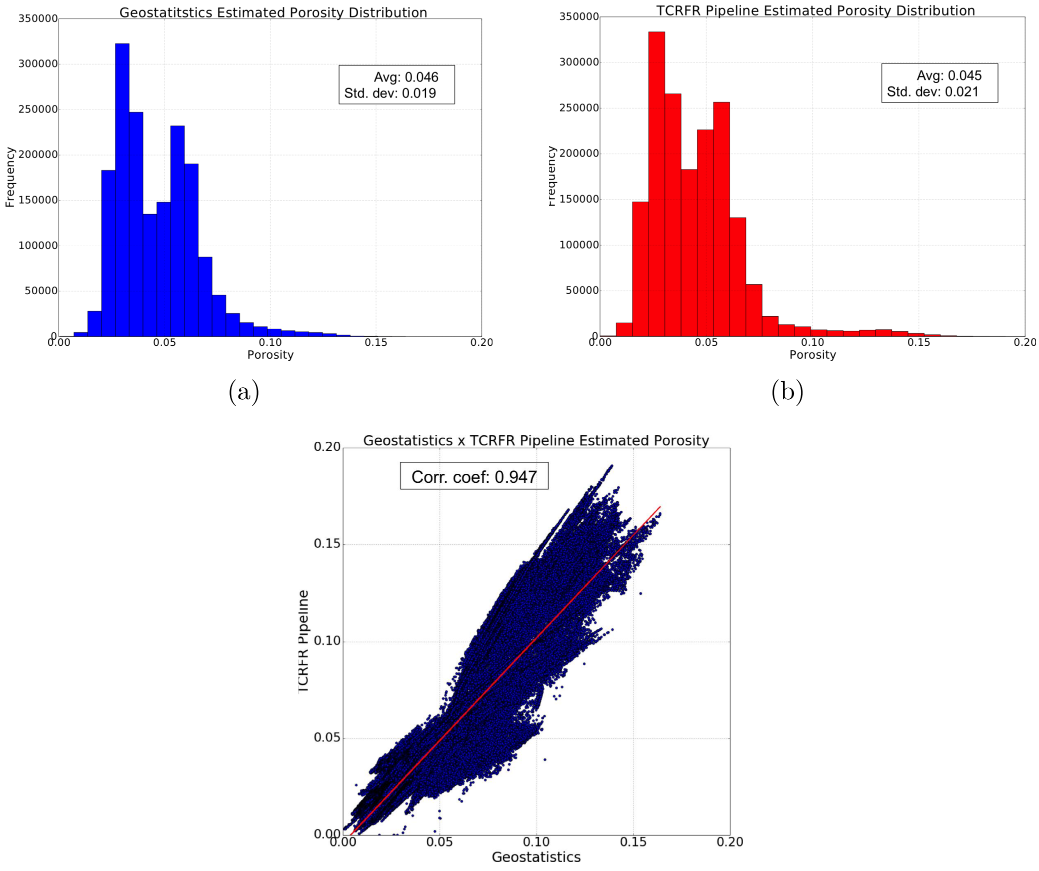
Table 6.5 shows the RMSE and MDAE errors by original TCRFR and TCRFR Pipeline from the geostatistics estimation and Figure 6.20 presents the estimated porosity histograms and a cross plot comparing the geostatistics and TCRFR Pipeline estimation results. Again, one can observe that the results from the improved TCRFR are similar to the ones obtained with the geostatistical approach.

Method	RMSE	MDAE
TCRFR	0.89065	0.48472
TCRFR Pipeline	0.44430	0.17930

Table 6.5: Comparison between TCRFR and TCRFR pipeline on the real dataset. Errors are evaluated by using the geostatistics estimation as reference.

It is worth noting that the TCRFR Pipeline gives sharper contours than the geostatistics estimation. Although one cannot argue with only the current results that this is an advantage of the TCRFR Pipeline over the geostatistics estimation, it might imply that the proposed semi-automatic method has a potential to improve even the geostatistics estimation.

In general, even with partial facies overlap, the TCRFR Pipeline is able to estimate the different facies present in a reservoir as long as the corresponding regression models are distinct, i.e., if the slope and/or intercept for each facies linear regressor is different from all the others.



(a)

(b)

(c)

Figure 6.20: Porosity statistics: (a) geostatistics porosity distribution; (b) TCRFR Pipeline porosity distribution; (c) geostatistics vs. TCRFR Pipeline porosity cross plot.

7
Conclusions

This work tackled the problem of porosity prediction in petroleum reservoirs, a fundamental task in the oil industry.

Handling data under spatial structures with limited number of labels remains a great challenge, requiring novel and robust modeling strategies. In this particular case, porosity needs to be predicted in the whole reservoir from local, labeled and accurate porosity information in the wells combined with plentiful but imprecise impedance input information available everywhere in the reservoir volume.

The proposed Transductive Conditional Random Field (TCRFR) is an automatic statistical inference method able to estimate hidden or *latent* states of geological facies, propagating information in the reservoir based on a conditional random field (CRF) probabilistic graphical model and generating multiple regression models dependent on the estimated facies to predict the corresponding porosities. This way, both facies and porosities of the unlabeled samples are simultaneously estimated, respecting the inferred spatial structure learned from the conditional random field. The method implement with concepts like semi-supervised/transductive learning, classification based on non-i.i.d. data with spatial dependency structure, latent state inference, and regression from continuous labels.

To handle extreme but realistic scenarios, where only a scarce number of porosity samples from a few exploratory wells are available in the reservoir, two new preprocessing techniques were also proposed, inspired in the image processing literature. The first technique performs a graph-based 3D volume segmentation, while the second one makes use of label annotation of facies. The whole workflow, considering both preprocessing techniques plus TCRFR is called the TCRFR Pipeline.

Experiments on both synthetic and real-world datasets were conducted, first analyzing the TCRFR performance alone and then the TCRFR Pipeline performance. In both cases TCRFR and TCRFR Pipeline presented superior performance when compared with state-of-the-art competitors, as well as with the traditional geostatistics approach.

Execution time was approximately linear with increasing number of samples. Sensitivity analysis showed a remarkable robustness of the TCRFR Pipeline to noise in the seismic impedance input.

As future work, code parallelization needs to be implemented for scalability. Also, domain knowledge can be added to the current architecture. Last but no least, the method can be applied to other fields that also present spatial and/or temporal dependency structure.

References

[1] GEOMORE. Oil On My Shoes – The Original Petroleum Geology Site. http://www.geomore.com/porosity-and-permeability-2, 2016.

[2] SCHLUMBERGER. The Oilfield Glossary. http://www.glossary.oilfield.slb.com/Terms/s/seismic_acquisition.aspx, 2016.

[3] WIKIPEDIA. Geology of the Capital Reef area. https://en.wikipedia.org/wiki/Geology_of_the_Capitol_Reef_area, 2015.

[4] CASTRO, S.; CAERS, J.; MUKERJI, T. The Stanford VI reservoir. *18th Annual Report. Stanford Center for Reservoir Forecasting (SCRF)*, p. 1–73, 2005.

[5] JOHNSON, C. Connor Johnson Blog about math, programming, and data. http://connor-johnson.com/2014/03/20/simple-kriging-in-python, 2014.

[6] BISHOP, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006. v. 4 of *Information science and statistics*.

[7] KOLLER, D.; FRIEDMAN, N. *Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning series)*. 1st.. ed. The MIT Press, 2009.

[8] SCHLUMBERGER. The Oilfield Glossary. http://www.glossary.oilfield.slb.com/Terms/p/porosity.aspx, 2015.

[9] CHAPELLE, O.; SCHÖLKOPF, B.; ZIEN, A. *Semi-Supervised Learning*. 1st. ed. The MIT Press, 2010.

[10] ZHU, X.; GOLDBERG, A. B. *Introduction to Semi-Supervised Learning*. The MIT Press, 2009. v. 3.

[11] VAPNIK, V. *Statistical Learning Theory*. First. ed. Wiley, 1998.

[12] DOYEN, P. M. Porosity from seismic data: A geostatistical approach. *Geophysics*, v. 53, n. 10, p. 1263, 1988.

[13] DEUTSCH, C. V.; JOURNEL, A. G. *GSLIB - Geostatistical Software Library and User's Guide*. 2nd. ed. Oxford University Press, 1998.

[14] DUBRULE, O. *Geostatistics for Seismic Data Integration in Earth Models*. SEG, 2003.

[15] CAERS, J. *Petroleum Geostatistics*. Society of Petroleum Engineers, 2005.

[16] LARSEN, A.; ULVMOEN, M.; OMRE, H.; BULAND, A. Bayesian lithology/fluid prediction and simulation on the basis of a Markov-chain prior model. *Geophysics*, v. 71, n. 5, p. R69–R78, 2006.

[17] MUKERJI, T.; AVSETH, P.; MAVKO, G.; TAKAHASHI, I. Statistical rock physics: Combining rock physics, information theory, and geostatistics to reduce uncertainty in seismic reservoir characterization. *The Leading Edge*, v. 20, n. 3, p. 313–319, 2001.

[18] DOYEN, P. M. *Seismic Reservoir Characterization: An Earth Modeling Perspective*. EAGE Publications, 2007.

[19] MAVKO, G.; MUKERJI, T.; DVORKIN, J. *The Rock Physics Handbook*. 2nd.. ed. Cambridge University Press, 2009.

[20] AVSETH, P.; MUKERJI, T.; MAVKO, G. *Quantitative Seismic Interpretation*. Cambridge University Press, 2010.

[21] MUKERJI, T.; JØRSTAD, A.; AVSETH, P.; MAVKO, G.; GRANLI, J. R. Mapping lithofacies and pore-fluid probabilities in a North Sea reservoir: Seismic inversions and statistical rock physics. *Geophysics*, v. 66, p. 988–1001, 2001.

[22] GUNNING, J.; GLINSKY, M. Delivery: an open-source model-based Bayesian seismic inversion program. *Computers & Geosciences*, v. 30, p. 619–636, 2004.

[23] EIDSVIK, J.; MUKERJI, T.; SWITZER, P. Estimation of Geological Attributes from a Well Log: An Application of Hidden Markov Chains. *Mathematical Geology*, v. 36, n. 3, p. 379–397, apr 2004.

[24] SPIKES, K.; MUKERJI, T.; DVORKIN, J.; MAVKO, G. Probabilistic seismic inversion based on rock-physics models. *Geophysics*, v. 72, n. 5, p. R87–R97, 2007.

[25] GRANA, D.; ROSSA, E. D. Probabilistic petrophysical-properties estimation integrating statistical rock ohysics with seismic inversion. *Geophysics*, v. 75, n. 3, p. O21–O37, 2010.

[26] EIDSVIK, J.; OMRE, H.; MUKERJI, T.; MAVKO, G.; AVSETH, P. Seismic reservoir prediction using Bayesian integration of rock physics and Markov random fields; a North Sea example. *The Leading Edge*, v. 21, p. 290–294, 2002.

[27] SAMS, M.; ATKINS, D.; SAID, P.; PARWITO, E.; VAN RIEL, P. Stochastic Inversion for High Resolution Reservoir Characterisation in the Central Sumatra Basin. In: . c1999.

[28] AL-ANAZI, A. F.; GATES, I. D. Support vector regression for porosity prediction in a heterogeneous reservoir: A comparative study. *Computers & Geosciences*, v. 36, n. 12, p. 1494–1503, 2010.

[29] LEITE, E. P.; VIDAL, A. C. 3D Porosity prediction from seismic inversion and neural networks. *Computers & Geosciences*, v. 37, n. 8, p. 1174–1180, 2011.

[30] DUBOIS, M. K.; BOHLING, G. C.; CHAKRABARTI, S. Comparison of four approaches to a rock facies classification problem. *Computers and Geosciences*, v. 33, n. 5, p. 599–617, 2007.

[31] CRACKNELL, M. J.; READING, A. M. Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information. *Computers & Geosciences*, v. 63, p. 22–33, 2014.

[32] LAFFERTY, J.; MCCALLUM, A. Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data. *International Conference on Machine Learning (ICML)*, v. 2001, p. 282–289, 2001.

[33] PENG, J.; BO, L.; XU, J. Conditional Neural Fields. *Advances in Neural Information Processing Systems*, v. 9, p. 1–9, 2009.

[34] ROBINSON, P.; MORENCY, L.-P. Continuous Conditional Neural Fields for Structured Regression. , n. 1, p. 1–16, 2014.

[35] BALTRUSAITIS, T.; BANDA, N.; ROBINSON, P. Dimensional Affect Recognition using Continuous Conditional Random Fields. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, p. 1–8, 2013.

[36] TSOCHANTARIDIS, I.; HOFMANN, T. Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Researchnal of Machine Learning Research*, v. 6, p. 1453–1484, 2005.

[37] RATLIFF, N. D.; BAGNELL, J. A.; ZINKEVICH, M. A. (Online) Subgradient Methods for Structured Prediction. *Artificial Intelligence and Statistics*, v. 2007, 2007.

[38] BO, L.; SMINCHISESCU, C. Structured Output-Associative Regression. In: . c2009. p. 2403–2410.

[39] BLASCHKO, M. B.; LAMPERT, C. H. Learning to localize objects with structured output regression. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 5302 LNCS, n. PART 1, p. 2–15, 2008.

[40] CORTES, C.; MOHRI, M. On Transductive Regression. *Advances in Neural Information Processing Systems 19*, p. 305–312, 2007.

[41] CHAPELLE, O.; VAPNIK, V.; WESTON, J. Transductive Inference for Estimating Values of Functions. *Advances in Neural Information Processing Systems (NIPS)*, v. 12, n. 5, p. 421 – 427, 1999.

[42] SINDHWANI, V.; NIYOGI, P.; BELKIN, M. Beyond the point cloud: from transductive to semi-supervised learning. In: . c2005. v. 1. p. 824–831.

[43] MELACCI, S.; BELKIN, M. Laplacian Support Vector Machines Trained in the Primal. *Journal of Machine Learning Research*, v. 12, p. 1149–1184, 2011.

[44] BELKIN, M.; NIYOGI, P.; SINDHWANI, V. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research*, v. 7, p. 2399–2434, 2006.

[45] JACOBS, R. A.; JORDAN, M. I. Adaptive Mixtures of Local Experts. *Neural Computation*, v. 3, p. 79–87, 1991.

[46] PAWELZIK, K.; KOHLMORGEN, J.; MÜLLER, K.-R. Annealed Competition of Experts for a Segmentation and Classification of Switching Dynamics. *Neural Computation*, v. 8, p. 340–356, 1996.

[47] LEISCH, F. FlexMix : A General Framework for Finite Mixture Models and Latent Class Regression in R. *Journal of Statistical Software*, v. 11, n. 8, 2004.

[48] GRÜN, B.; LEISCH, F. FlexMix Version 2 : Finite Mixtures with Concomitant Variables and Varying and Constant Parameters. *Journal of Statistical Software*, v. 28, n. 4, 2008.

[49] RUSU, C.; RUSU, V. Radial Basis Functions Versus Geostatistics in Spatial Interpolations. *IFIP International Federation for Information Processing*, v. 217, n. 1, p. 119–128, 2006.

[50] MATHERON, G. *Traité de geoestatistique apliquée*. Paris: Editions Technip, 1962. v. 1.

[51] DEUTSCH, C. V. *Geostatistical Reservoir Modeling*. 1st. ed. Oxford University Press, 2002.

[52] SOARES, A. *Geostatística para as Ciências da Terra e do Ambiente*. 2nd. ed. IST Press, 2006.

[53] WIKIPEDIA. Kriging. https://en.wikipedia.org/wiki/Kriging, 2016.

[54] XU, W.; TRAN, T.; SRIVASTAVA, R.; JOURNEL, A. G. Integrating Seismic Data in Reservoir Modeling: The Collocated Cokriging Alternative. In: . c1992. p. 833–842.

[55] GOOVAERTS, P. Comparative performance of indicator algorithms for modeling conditional probability distribution functions. *Math Geology*, v. 26, n. 3, p. 385–410, 1994.

[56] SUTTON, C. An Introduction to Conditional Random Fields. *Foundations and Trends® in Machine Learning*, v. 4, n. 4, p. 267–373, 2012.

[57] HE, X.; ZEMEL, R.; CARREIRA-PERPINAN, M. Multiscale Conditional Random Fields for Image Labeling. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, v. 2, p. 695 –702, 2004.

[58] TASKAR, B.; KLEIN, D.; COLLINS, M.; KOLLER, D.; MANNING, C. Max-Margin Parsing. In: . c2004.

[59] BERNAL, A.; CRAMMER, K.; HATZIGEORGIOU, A.; PEREIRA, F. Global Discriminative Learning for Higher- Accuracy Computational Gene Prediction. *PLoS Computational Biology*, v. 3, n. 3, p. 488–497, 2007.

[60] WALLACH, H. M. Conditional Random Fields: An Introduction. *Department of Computer and Information Science, University of Pennsylvania - Technical Report MS-CIS-04-21*, 2004.

[61] KLINGER, R.; TOMANEK, K. Classical Probabilistic Models and Conditional Random Fields. Technical Report December, Department of Computer Science, Dortmund University of Technology, 2007.

[62] MARKOV, A. *The Theory of Algorithms*. Academy of Sciences of the USSR, 1954.

[63] HAMMERSLEY, J.; CLIFFORD, P. Markov fields on finite graphs and lattices. 1971.

[64] BESAG, J. Statistical Analysis of Non-Lattice Data. *The Statistician*, v. 24, n. 3, p. 179–195, 1974.

[65] ELKAN, C. Log-linear models and conditional random fields. *Tutorial notes at CIKM*, v. 8, 2008.

[66] LIU, D. C.; NOCEDAL, J. On The Limited Memory BFGS Method for Large Scale Optimization. *Mathematical Programming*, v. 45, p. 503–528, 1989.

[67] PEARL, J. Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach. *AAAI Proceedings*, p. 133–136, 1982.

[68] LAURITZEN, S. L.; SPIEGELHALTER, D. J. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, v. 50, n. 2, p. 157–224, 1988.

[69] YEDIDIA, J. S.; FREEMAN, W. T.; WEISS, Y. Understanding Belief Propagation and its Generalizations. *Intelligence*, v. 8, p. 236–239, 2002.

[70] MÜLLER, K. R.; MIKA, S.; RÄTSCH, G.; TSUDA, K.; SCHÖLKOPF, B. An Introduction to Kernel-Based Learning Algorithms, 2001.

[71] YUILLE, A. L.; RANGARAJAN, A. The Concave-Convex Procedure (CCCP). *Neural computation*, v. 15, n. 4, p. 915–36, 2003.

[72] AN, L. T. H.; TAO, P. D. The DC (Difference of Convex Functions) Programming and DCA Revisited with DC Models of Real World Nonconvex Optimization Problems. *Annals of Operations Research*, v. 133, n. 1-4, p. 23–46, 2005.

[73] YU, C.-N. J.; JOACHIMS, T. Learning structural SVMs with latent variables. *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, New York, New York, USA, p. 1–8, 2009.

[74] WAINWRIGHT, M. J.; JORDAN, M. I. Variational inference in graphical models: The view from the marginal polytope. *Graphical Models*, v. 41, n. 2, p. 961—-971, 2003.

[75] WEISS, Y.; FREEMAN, W. T. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural computation*, v. 13, n. 10, p. 2173–2200, 2001.

[76] FELZENSZWALB, P. F.; HUTTENLOCHER, D. P. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, v. 59(2), p. 167–181, 2004.

[77] WALTHER, J. Einleitung in die Geologie als historische Wissenschaft. *Lithogenesis der Gegenwart*, v. 3, p. 535–1055, 1894.

[78] MIALL, A. D. *Fluvial Depositional Systems*. Springer Geology, 2014.

[79] DVORKIN, J.; GUTIERREZ, M. A.; GRANA, D. *Seismic Reflections of Rock Properties*. 1st.. ed. Cambridge University Press, 2014.

[80] WIKIPEDIA. Tikhonov regularization. https://en.wikipedia.org/wiki/Tikhonov_regularization, 2016.

[81] DRUCKER, H.; BURGES, C. J. C.; KAUFMAN, L.; SMOLA, A. J.; VAPNIK, V. N. Support Vector Regression Machines. *Advances in Neural Information Processing Systems*, v. 9, p. 155–161, 1996.

[82] STATLEC. Statlec - Fundamentals of probability. https://www.statlect.com/fundamentals-of-probability/indicator-functions, 2016.

[83] RAND, W. M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, v. 66, p. 846–850, 1971.

[84] HUBERT, L.; ARABIE, P. Comparing Partitions. *Journal of Classification*, v. 2, n. 1, p. 193–218, 1985.

A
Miscellaneous Concepts and Definitions

A.1
Operations on Factors

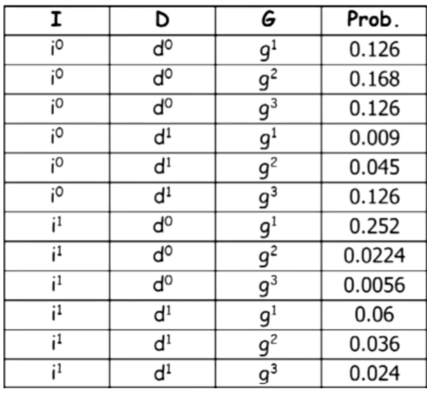
Factors are the fundamental building blocks in the definition and manipulation of probability distributions in probabilistic graphical models. In this section, we give some examples of their functionality. These examples were extracted from [7].

A factor is a function that takes a set of arguments that represent random variables and returns some real value:

$$f(\boldsymbol{x}_1, ...\boldsymbol{x}_k) \rightarrow R \in \mathbb{R}$$

So, factor f gets all combinations of the variables $(\boldsymbol{x}_1, ...\boldsymbol{x}_k)$ in its scope and provides a R output for each combination.

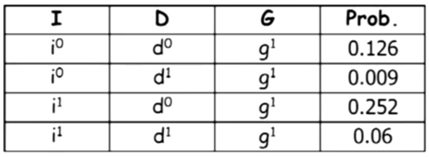
A joint distribution, for instance, is a factor. Let us take the $P(I, D, G)$ distribution in Figure A.1. Any combination of variables I, D, and G gives a number, which in this particular case is a probability and all probabilities sum up to 1.



I	D	G	Prob.
i^0	d^0	g^1	0.126
i^0	d^0	g^2	0.168
i^0	d^0	g^3	0.126
i^0	d^1	g^1	0.009
i^0	d^1	g^2	0.045
i^0	d^1	g^3	0.126
i^1	d^0	g^1	0.252
i^1	d^0	g^2	0.0224
i^1	d^0	g^3	0.0056
i^1	d^1	g^1	0.06
i^1	d^1	g^2	0.036
i^1	d^1	g^3	0.024

Figure A.1: Example of factor of a joint distribution [7].

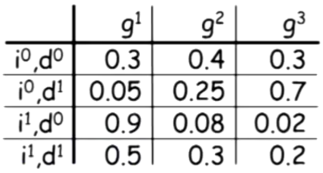
It is important to keep in mind that factors do not need to be normalized measures. In Figure A.2, we fix $G = g^1$ and in this case the scope of the factor is $f(I, D)$.



I	D	G	Prob.
i^0	d^0	g^1	0.126
i^0	d^1	g^1	0.009
i^1	d^0	g^1	0.252
i^1	d^1	g^1	0.06

Figure A.2: Example of factor of an unnormalized measure [7].

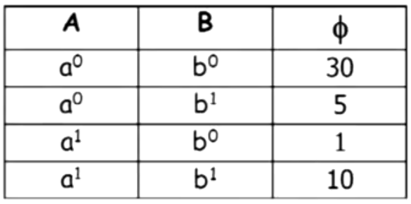
Another example is a factor that represents a conditional probability distribution $P(G|I, D)$, as shown in Figure A.3. In this case, variable G can assume one of three values $\{g^1, g^2, g^3\}$ and the probabilities of these values sum up to 1 horizontally on the table, depending on the values of I and D.



	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

Figure A.3: Example of factor of a conditional distribution [7].

We can have general factors, as the one shown in Figure A.4. We see that those factors are scores that are not constrained to the interval $[0, 1]$. The values assigned to variables A and B could be real numbers, for instance.



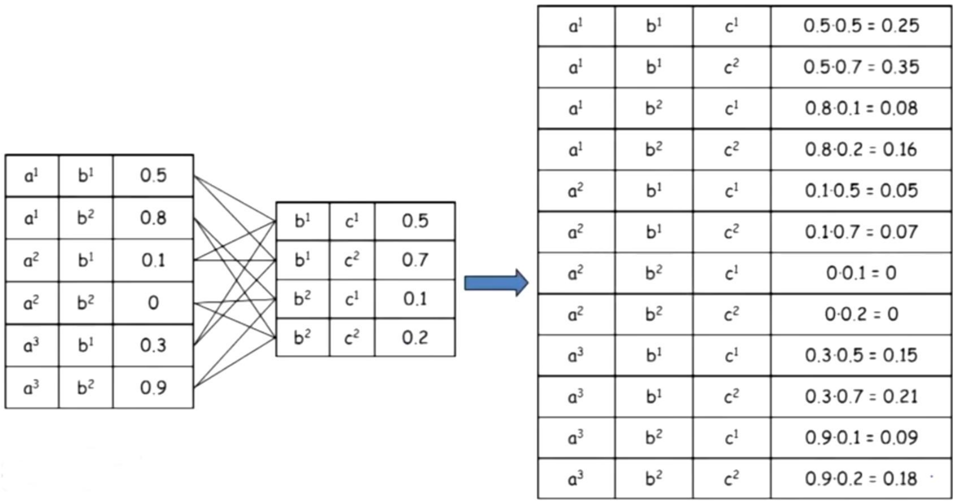
A	B	ϕ
a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

Figure A.4: Example of a general factor [7].

One of the most common operations on factors is the *factor product*, as we see in the example of Figure A.5. We take to factors, $f_1(A, B)$ and $f_2(B, C)$. The resulting function is going to combine all three original arguments, $f_3(A, B, C)$, and its values is simply the multiplication of the rows in first and second tables.

Another important operation is *factor marginalization*. In Figure A.6 we have a factor $f(A, B, c)$ and if we want to marginalize out B, we take both values for each combination of A and C values and add them up. This process is similar to probability marginalization, except that the values involved don't need to be probabilities.

Finally, we can also have *factor reduction*. In Figure A.7, we want to reduce a factor $f_1(A, B, C)$ to all occurrences of c^1. In the end, we obtain a factor $f_2(A, B, c^1) = f_2(A, B)$.

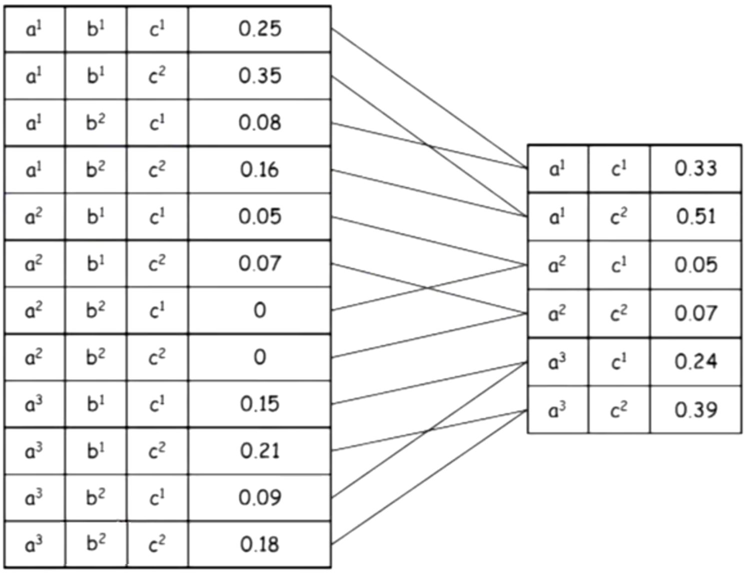


a^1	b^1	0.5
a^1	b^2	0.8
a^2	b^1	0.1
a^2	b^2	0
a^3	b^1	0.3
a^3	b^2	0.9

b^1	c^1	0.5
b^1	c^2	0.7
b^2	c^1	0.1
b^2	c^2	0.2

a^1	b^1	c^1	0.5·0.5 = 0.25
a^1	b^1	c^2	0.5·0.7 = 0.35
a^1	b^2	c^1	0.8·0.1 = 0.08
a^1	b^2	c^2	0.8·0.2 = 0.16
a^2	b^1	c^1	0.1·0.5 = 0.05
a^2	b^1	c^2	0.1·0.7 = 0.07
a^2	b^2	c^1	0·0.1 = 0
a^2	b^2	c^2	0·0.2 = 0
a^3	b^1	c^1	0.3·0.5 = 0.15
a^3	b^1	c^2	0.3·0.7 = 0.21
a^3	b^2	c^1	0.9·0.1 = 0.09
a^3	b^2	c^2	0.9·0.2 = 0.18

Figure A.5: Example of factor product [7].

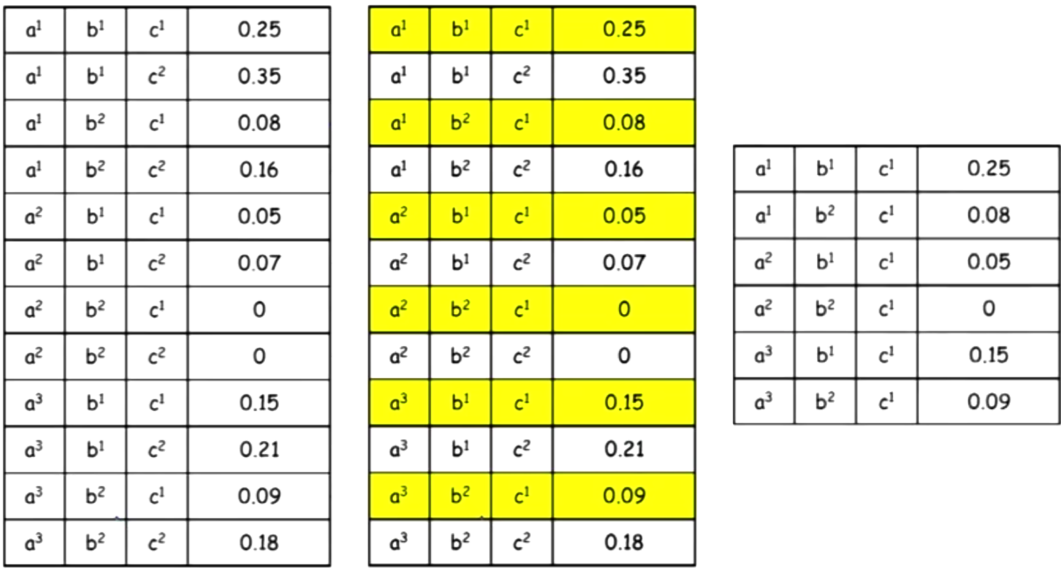


a^1	b^1	c^1	0.25
a^1	b^1	c^2	0.35
a^1	b^2	c^1	0.08
a^1	b^2	c^2	0.16
a^2	b^1	c^1	0.05
a^2	b^1	c^2	0.07
a^2	b^2	c^1	0
a^2	b^2	c^2	0
a^3	b^1	c^1	0.15
a^3	b^1	c^2	0.21
a^3	b^2	c^1	0.09
a^3	b^2	c^2	0.18

a^1	c^1	0.33
a^1	c^2	0.51
a^2	c^1	0.05
a^2	c^2	0.07
a^3	c^1	0.24
a^3	c^2	0.39

Figure A.6: Example of factor marginalization [7].

In summary, an exponentially large probability distribution of N random variables is defined by taking small factors and putting them together by multiplying them to define probability distributions in high dimension spaces.



a^1	b^1	c^1	0.25
a^1	b^1	c^2	0.35
a^1	b^2	c^1	0.08
a^1	b^2	c^2	0.16
a^2	b^1	c^1	0.05
a^2	b^1	c^2	0.07
a^2	b^2	c^1	0
a^2	b^2	c^2	0
a^3	b^1	c^1	0.15
a^3	b^1	c^2	0.21
a^3	b^2	c^1	0.09
a^3	b^2	c^2	0.18

a^1	b^1	c^1	0.25
a^1	b^1	c^2	0.35
a^1	b^2	c^1	0.08
a^1	b^2	c^2	0.16
a^2	b^1	c^1	0.05
a^2	b^1	c^2	0.07
a^2	b^2	c^1	0
a^2	b^2	c^2	0
a^3	b^1	c^1	0.15
a^3	b^1	c^2	0.21
a^3	b^2	c^1	0.09
a^3	b^2	c^2	0.18

a^1	b^1	c^1	0.25
a^1	b^2	c^1	0.08
a^2	b^1	c^1	0.05
a^2	b^2	c^1	0
a^3	b^1	c^1	0.15
a^3	b^2	c^1	0.09

Figure A.7: Example of factor reduction [7].

A.2 Joint Feature Map

In a multiclass classification problem, a compact and often common way to express multiple classes y and multiple features $\boldsymbol{x}$ is through joint feature maps representations.

As an example, set's assume that we want to represent K classes learning problems as one. We can define class-dependent feature maps Ψ_j in the following representation:

$$
\begin{aligned}
\boldsymbol{x}_i &\mapsto (\Psi(\boldsymbol{x}), 0, 0, ..., 0) \equiv \Psi_1(\boldsymbol{x}_i), \quad if\ y_i = c_1 \\
\boldsymbol{x}_i &\mapsto (0, \Psi(\boldsymbol{x}), 0, ..., 0) \equiv \Psi_2(\boldsymbol{x}_i), \quad if\ y_i = c_2 \\
\vdots & \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
\boldsymbol{x}_i &\mapsto (0, 0, 0, ..., \Psi(\boldsymbol{x})) \equiv \Psi_K(\boldsymbol{x}_i), \quad if\ y_i = c_K
\end{aligned}
$$

For each class c_j we have a distinct set of weights w_j, so the joint weight vector is $\boldsymbol{w}_{joint} = (\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_K)$. In the same manner, the joint Hilbert spaces are defined as $\mathcal{H}_{joint} := \oplus_{j=1}^{K} \mathcal{H}_{\boldsymbol{x}}$ and $\boldsymbol{w}_j \in \mathcal{H}_{\boldsymbol{x}}^{j}$.

As a consequence, these two formulations are equivalent:

$$\langle \Psi(\boldsymbol{w}_j, \boldsymbol{x}_i) \rangle_{\mathcal{H}} \equiv \langle \Psi_j(\boldsymbol{w}_{joint}, \boldsymbol{x}_i) \rangle_{\mathcal{H}_{joint}}$$

At this point, we have defined one feature map Ψ_j for each output class

$c_j \in \mathcal{Y}$:

$$\Psi_j : \mathcal{X} \rightarrow \mathcal{H}_{joint}$$

From there, we can define just one **joint** feature map Φ, which depends on sample $\boldsymbol{x}$ and on the class label y:

$$\begin{aligned} \Phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{H}_{joint} \\ \Phi(\boldsymbol{x}, y) = \Psi_j(\boldsymbol{x}) \; for \; y = c_j \end{aligned} \tag{A-1}$$

Equation A-1 can be expressed theoretically as the following matrix:

$$\Phi(\boldsymbol{x}, y) = \begin{array}{c} \begin{array}{ccccc} \Psi_1(\boldsymbol{x},1) & \Psi_2(\boldsymbol{x},2) & \Psi_3(\boldsymbol{x},3) & \dots & \Psi_K(\boldsymbol{x},K) \end{array} \\ \begin{bmatrix} \boldsymbol{x} & 0 & 0 & \dots & 0 \\ 0 & \boldsymbol{x} & 0 & \dots & 0 \\ 0 & 0 & \boldsymbol{x} & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & \boldsymbol{x} \end{bmatrix} \end{array}$$

$$= \begin{bmatrix} \sum_{i=1}^{m} f_1(\boldsymbol{x}, y) & 0 & 0 & \dots & 0 \\ 0 & \sum_{i=1}^{m} f_2(\boldsymbol{x}, y) & 0 & \dots & 0 \\ 0 & 0 & \sum_{i=1}^{m} f_3(\boldsymbol{x}, y) & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & \sum_{i=1}^{m} f_K(\boldsymbol{x}, y) \end{bmatrix},$$

and considering a objective function $\mathcal{L}(\boldsymbol{x}) = \text{argmax}_y \langle \boldsymbol{w}_{joint}^T, \Phi(\boldsymbol{x}, y) \rangle$, the cross product can be written as

$$\begin{bmatrix} \sum_{i=1}^{m} f_1(\boldsymbol{x}, y) & 0 & 0 & \dots & 0 \\ 0 & \sum_{i=1}^{m} f_2(\boldsymbol{x}, y) & 0 & \dots & 0 \\ 0 & 0 & \sum_{i=1}^{m} f_3(\boldsymbol{x}, y) & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & \sum_{i=1}^{m} f_K(\boldsymbol{x}, y) \end{bmatrix} \bullet \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \boldsymbol{w}_3 \\ \vdots \\ \boldsymbol{w}_K \end{bmatrix}$$

A.3
Indicator Function

The indicator function of some event (or condition) is a random variable that takes value 1 when that event happens and value 0 otherwise. Indicator functions are often used in probability theory to simplify notation and to prove theorems [82].

Formally, we can define an indicator function as

$$1_E(\omega) = \begin{cases} 1 & \text{if } \omega \in E \\ 0 & \text{if } \omega \notin E \end{cases},$$

where ω is the random variable and E represents the event.

Besides $1_E(\omega)$ notation, the indicator function is also often written as $1[\omega \in E]$.

As an example, an indicator function with the condition $y = 1$ and $x = 0$ would be represented as

$$1[y = 1 \wedge x = 0] = \begin{cases} 1 & \text{if } y = 1 \text{ and } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

Another example would be the toss of a die. All the possible outputs are in the set $\Omega = \{1, 2, 3, 4, 5, 6\}$. The indicator function that defines any even output number is $1[\omega = 2 \vee \omega = 4 \vee \omega = 6]$.

A.4 Adjusted Rand Index

The Rand Index [83] computes the similarity measure between two clusters by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and the true clusters.

Given a set $S = \{o_1, o_2, ..., o_n\}$ with n elements, let us define two distinct partitions of the objects in S, $T = \{t_1, t_2, ..., t_r\}$ and $P = \{p_1, p_2, ..., p_s\}$, such that $\cup_{i=1}^{r} t_i = \cup_{j=1}^{s} p_j = S$ and $t_i \cap t_{i'} = p_j \cap p_{j'} = 0$ for $1 \leq i \neq i' \leq r$ and $1 \leq j \neq j' \leq s$.

Let us now consider the following groups:

- ***a***, the number of pairs of objects in S that are in the same set in T and in the same set in P;
- ***b***, the number of pairs of objects in S that are in different sets in T and in different sets in P;
- ***c***, the number of pairs of objects in S that are in the same set in T and in different sets in P;
- ***d***, the number of pairs of objects in S that are in different sets in T and in the same set in P.

True\Predicted	p_1	p_2	$\dots$	p_s	Sums
t_1	n_{11}	n_{12}	$\dots$	n_{1s}	$n_{1.}$
t_2	n_{21}	n_{22}	$\dots$	n_{2s}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
t_r	n_{r1}	n_{r2}	$\dots$	n_{rs}	$n_{r.}$
Sums	$n_{.1}$	$n_{.2}$	$\dots$	$n_{.s}$	n

Table A.1: Notation for comparing two clusters.

The Rand Index R is defined as

$$RandIndex = \frac{\boldsymbol{a}+\boldsymbol{b}}{\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c}+\boldsymbol{d}},$$

where intuitively $\boldsymbol{a}+\boldsymbol{b}$ represents the number of agreements between C_1 and C_2 and $\boldsymbol{c}+\boldsymbol{d}$ indicates the number of disagreements between C_1 and C_2.

The Rand Index tends to give quite large values even when clustering methods are in substantial disagreement. Even a random assignment of points to clusters can lead to large Rand Index values. Hubert and Arabie [84] proposed an adjustment to the Rand Index in order to account for agreement by chance. The true and and predicted clustering are selected at random so that the number of objects in both clustering is fixed.

The Adjusted Rand Index is obtained using

$$ARI = \frac{RandIndex - ExpectedRandIndex}{MaxRandIndex - ExpectedRandIndex}$$

It is shown in [84] that the Adjusted Rand Index can be written as

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}/\binom{n}{2}\right]}{\frac{1}{2}\left[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}\right] - \left[\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}/\binom{n}{2}\right]}, \tag{A-2}$$

with $\binom{x}{2} = \frac{x(x-1)}{2}$.

In Equation A-2, n_{ij} represents the number of objects that are in clusters t_i and s_j, $n_{i.}$ is is the number of objects that are in cluster t_i, and $n_{.j}$ is the number of objects in cluster p_j, as shown in Table A.1.