2018-01-01

A Machine Learning Approach To Rendering 3d Stratigraphic Models Of The Hueco Bolson, Western Texas And Northern Mexico

Joel Gerardo Castro
University of Texas at El Paso, joey-92@live.com

A MACHINE LEARNING APPROACH TO RENDERING 3D STRATIGRAPHIC MODELS OF THE HUECO BOLSON, WESTERN TEXAS AND NORTHERN MEXICO

JOEL G CASTRO II

Master's Program in Computational Science

APPROVED:

Diane Doser, Ph.D., Chair

Aaron A. Velasco, Ph.D.

Deana D. Pennington, Ph.D.

Suarav Kumar, Ph.D.

Charles Ambler, Ph.D.
Dean of the Graduate School

Dedication

Dedicated to my mother Gregoria Castro who taught me the true meaning of strength and perseverance, and to my father Joel Castro whom I still miss every day.

A MACHINE LEARNING APPROACH TO RENDERING 3D STRATIGRAPHIC MODELS OF THE HUECO BOLSON, WESTERN TEXAS AND NORTHERN MEXICO

by

JOEL G CASTRO II, BS

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Computational Science Program

THE UNIVERSITY OF TEXAS AT EL PASO

December 2018

Acknowledgements

I would like to thank my advisor Dr. Diane Doser for her support. She is a remarkable scientist and teacher who has inspired me to strive for greatness. I would like to thank my committee; Dr. Pennington, Dr. Velasco, and Dr. Kumar for helping me develop my ideas, and helping me prepare for my defense.

Finally, I would like to acknowledge with gratitude my parents: Gregoria, and Lorenzo, my brother: Adriel, my girlfriend: Brenda, and my friends: Kat, Victor, Tony, Sandy, Kameron, and Jen for their unwavering love, and continuous encouragement throughout my years of studying, researching, and writing this thesis. Thank you.

Abstract

Data visualization is an effective way to analyze large amounts of spatial information to identify correlations, trends, outliers, and patterns. In this thesis I test the application of supervised machine learning algorithms to render a series of 3D visualizations designed to highlight general traits of the Hueco Bolson, a geologic basin located east of the Franklin Mountains in far west Texas and northern Mexico. A geologic basin is one of the most common inland places where sediments are collected. The geology of basins is of much interest to geophysicists, hydrologists, paleontologists, and oil prospectors. Here the task of 3D geologic modeling is approached as a classification problem. The 3D models constructed from this study are built from interpretation data (geologic cross sections) developed from previous studies conducted on the basin.

This new approach to geomodeling addresses some of the limitations associated with surface based modeling in densely faulted areas (using traditional 3D interpolation schemes) and volume based modeling. The 3D models produced for this thesis have given geoscientists a general understanding of the geometry and structure of Hueco Bolson, which is the principal aquifer for the Greater El Paso Region. The basin is positioned in southwestern Texas and south-central New Mexico on the U.S./Mexican border. El Paso and its surrounding urban area currently relies on groundwater for over half of its water supply, and Ciudad Juarez relies entirely on groundwater from the Hueco Bolson aquifer, supporting the 2.5 million inhabitants of the region.

Table of Contents

List of Figures

Chapter 1: Introduction

Current approaches to generating 3D geologic models involve the reconstruction of individual geologic contacts as surfaces. These surfaces are 3D interpolations (usually applying some variant of Kriging) of control points. However, these 3D surfaces are ineffective at capturing complex geology in scenarios such as complicated faulting. Multiple procedures have been developed in response to these situations. One popular approach involves slicing a volumetric model into multiple sub volumes (based on fault intersections), interpolating 3D surfaces within each sub volume, generating volumes between those surfaces, and finally, merging all sub volumes (Hampson et al. 2012).

These popular procedures, although effective, are time consuming, tedious, complex, and inefficient. Furthermore, current 3D interpolation algorithms produce irregular surfaces when the input data consist of a few sparse cross-sections, typically requiring the application of a smoothing filter. Finding an algorithm that simplifies the process of volume reconstruction from interpreted data (geologic cross-sections) is greatly desired by geoscientists. The objective of this study is to test several supervised learning algorithms that offer solutions to this problem. Here I tested: Support Vector Classification (SVC), and Probabilistic Neural Network Classification (PNNC).

Section 1 – Regional Setting

The Hueco Bolson (Figure 1) is about 320 kilometers long and 40 kilometers wide (Sheng et al. 2001). It is enclosed by the Organ, Franklin, Sierra de Juarez and Sierra del Presidio mountain ranges to the west and the Sacramento, Quitman, Malone, Sierra Blanca, and Finlay mountain ranges to the east (Sheng et al. 2001; Avila et al. 2016). The Hueco Bolson is hydraulically

connected to the Tularosa Basin to the north, and is part of the Hueco-Tularosa aquifer (Hibbs et al. 1997). A buried structural high, which serves as the divide between these two basins, lies 11 to 16 km north of the Texas New Mexico border (Sheng et al. 2001; Avila et al. 2016).

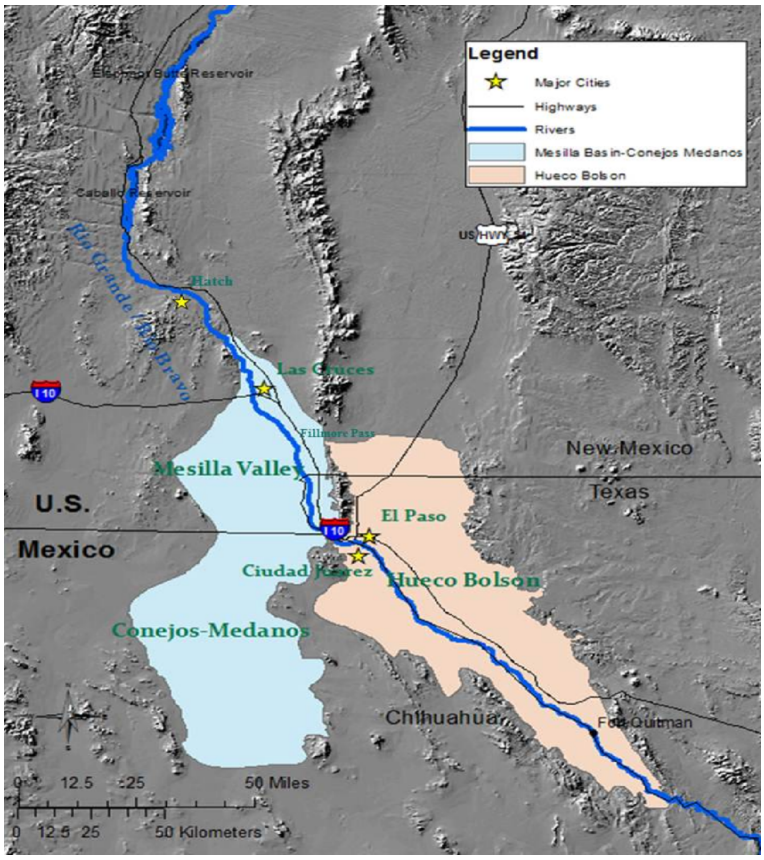


Figure 1:

Location of the Mesilla (blue) and Hueco (orange) Bolsons (Sheng et al. 2013).

Section 2 – Structure and Geologic History

The Hueco Bolson is a perfect example of a basin formed by extensional tectonics as a result of the Rio Grande Rift, one of the world's principle continental rift systems. This rift system extends from central Colorado, through New Mexico, to Presidio, Texas, and Chihuahua, Mexico (over 1000 km) (Olsen el al. 1987). In geology, a rift is a zone where the earth's crust and lithosphere are being pulled apart in opposite directions mainly due to convection currents

deep within the earth (Figure 2). Initiation of the Hueco Bolson rift basin began in the late Oligocene (33.9 to 29 million years ago) (Morgan et al. 1986) (see Figure 3 for geologic time scale), and maximum differential displacements that gave rise to the major Basin and Range structural blocks took place between the Late Miocene to the Late Pliocene (Figure 3) (23 to 1.8 million years ago) (Hawley et al. 2009). The array of normal faults found within the Hueco Bolson is a direct consequence of this tectonic extension.

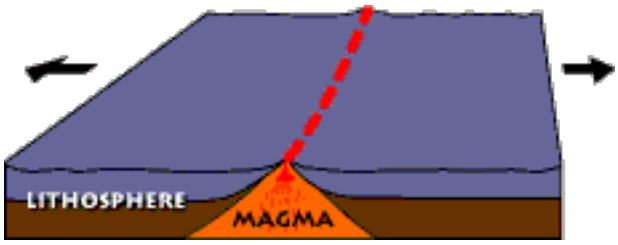
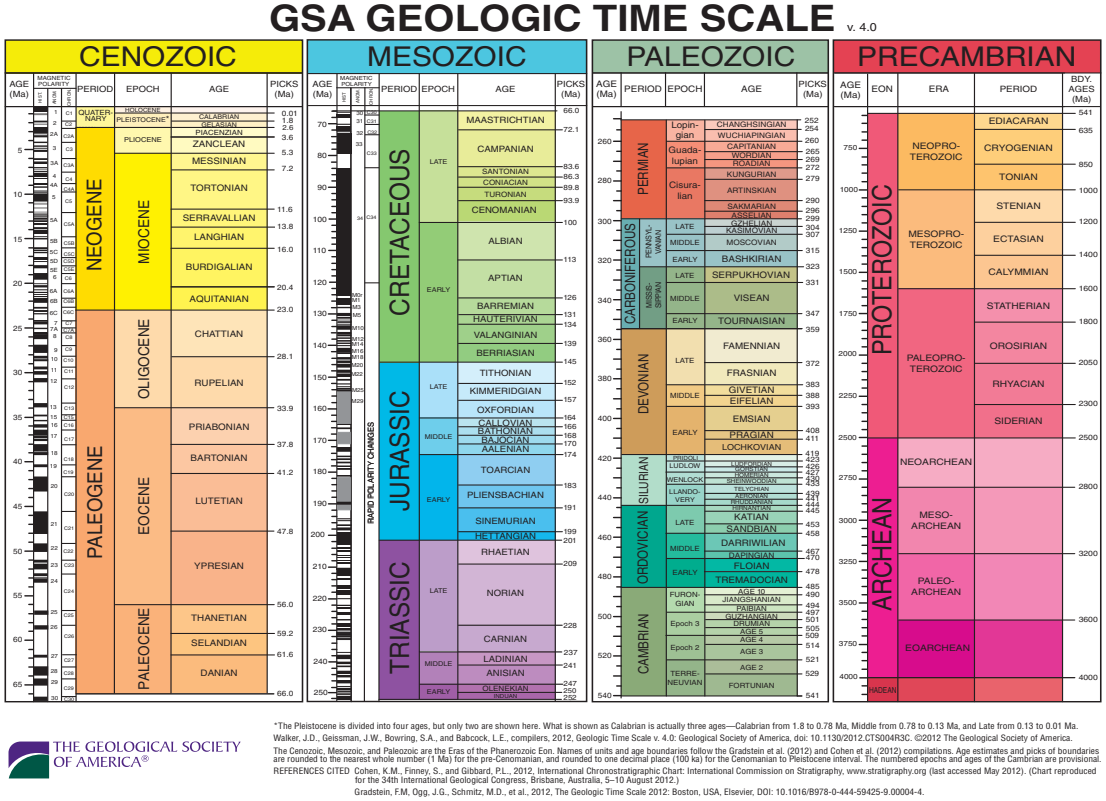


Figure 2:

Two tectonic plates moving apart from each other. This geologic event leads to the formation of a rift valley (https://geomaps.wr.usgs.gov/parks/pltec/diverge.html).



*The Pleistocene is divided into four ages, but only two are shown here. What is shown as Calabrian is actually three ages—Calabrian from 1.8 to 0.78 Ma, Middle from 0.78 to 0.13 Ma, and Late from 0.13 to 0.01 Ma.

Walker, J.D., Geissman, J.W., Bowring, S.A., and Babcock, L.E., compilers, 2012, Geologic Time Scale v. 4.0: Geological Society of America, doi: 10.1130/2012.CTS004R3C.

The Cenozoic, Mesozoic, and Paleozoic are the Eras of the Phanerozoic Eon. Names of units and age boundaries follow the Gradstein et al. (2012) and Cohen et al. (2012) compilations. Age estimates and picks of boundaries are rounded to the nearest whole number (1 Ma) for the pre-Cenomanian, and rounded to one decimal place (100 ka) for the Cenomanian to Pleistocene interval. The numbered epochs and ages of the Cambrian are provisional.

REFERENCES CITED Cohen, K.M., Finney, S., and Gibbard, P.L., 2012, International Chronostratigraphic Chart: International Commission on Stratigraphy, www.stratigraphy.org (last accessed May 2012). (Chart reproduced for the 34th International Geological Congress, Brisbane, Australia, 5–10 August 2012.)
Gradstein, F.M, Ogg, J.G., Schmitz, M.D., et al., 2012, The Geologic Time Scale 2012: Boston, USA, Elsevier, DOI: 10.1016/B978-0-444-59425-9.00004-4.

Figure 3

The Geological Society of America (GSA) geologic timescale illustrating the ages of different geologic units.

Section 3 – Bedrock

The bedrock units beneath the basin-fill deposits of the Hueco Bolson consist of various sedimentary and igneous rocks ranging in age from Precambrian to Lower Cenozoic (Figures 3 and 4) (1.1 billion to 40 million years ago). In the northern section of the Hueco Bolson we find Permian, Pennsylvania, Mississippian and Devonian strata, Silurian and Ordovician strata, and Precambrian strata (Figure 4) (George et al. 2011). In the southern section of the Hueco Bolson we find Cretaceous strata (Campogrande, Cox, Finley, and Bluff Mesa formations), and Permian strata (Figure 4) (George et al. 2011).

Most of the sedimentary strata beneath the basin-fill consists of carbonate rocks. Two major types of carbonate rocks are limestone, which is composed of calcite and aragonite ($CaCO_3$), and dolostone, which is composed of dolomite ($CaMg(CO_3)_2$). The remaining strata consist of sandstones, mud stones, and shale. Sandstones consist of quartz grains cemented together. Shales are soft, finely stratified rocks formed from consolidated mud or clay with parallel laminations. Mudstones are consolidated mud, but lack the laminations of shale.

The Cretaceous aged strata (Figure 4) in our study area consist of limestone, silty to shaly limestone, and shale. The Permian aged strata are primarily limestone, sandstone, and red-bed mudstones. The Pennsylvania, Lower Permian, and Upper Paleozoic aged strata (Figure 4) are made up of limestone and sandstone redbeds, sandy mudstone, with sections of shale, sandstone, and gypsite (primarily ($CaSO_4.2H_2O$)). The Middle Paleozoic (Devonian and Mississippian) and Lower Paleozoic (Cambrian and Silurian) strata (Figure 4) consist primarily of carbonate types with some shale (Hawley et al. 2009). The Precambrian aged strata come in the form of igneous and metamorphic rocks that are over 1 billion years old (Urbanczyk et al. 2001).

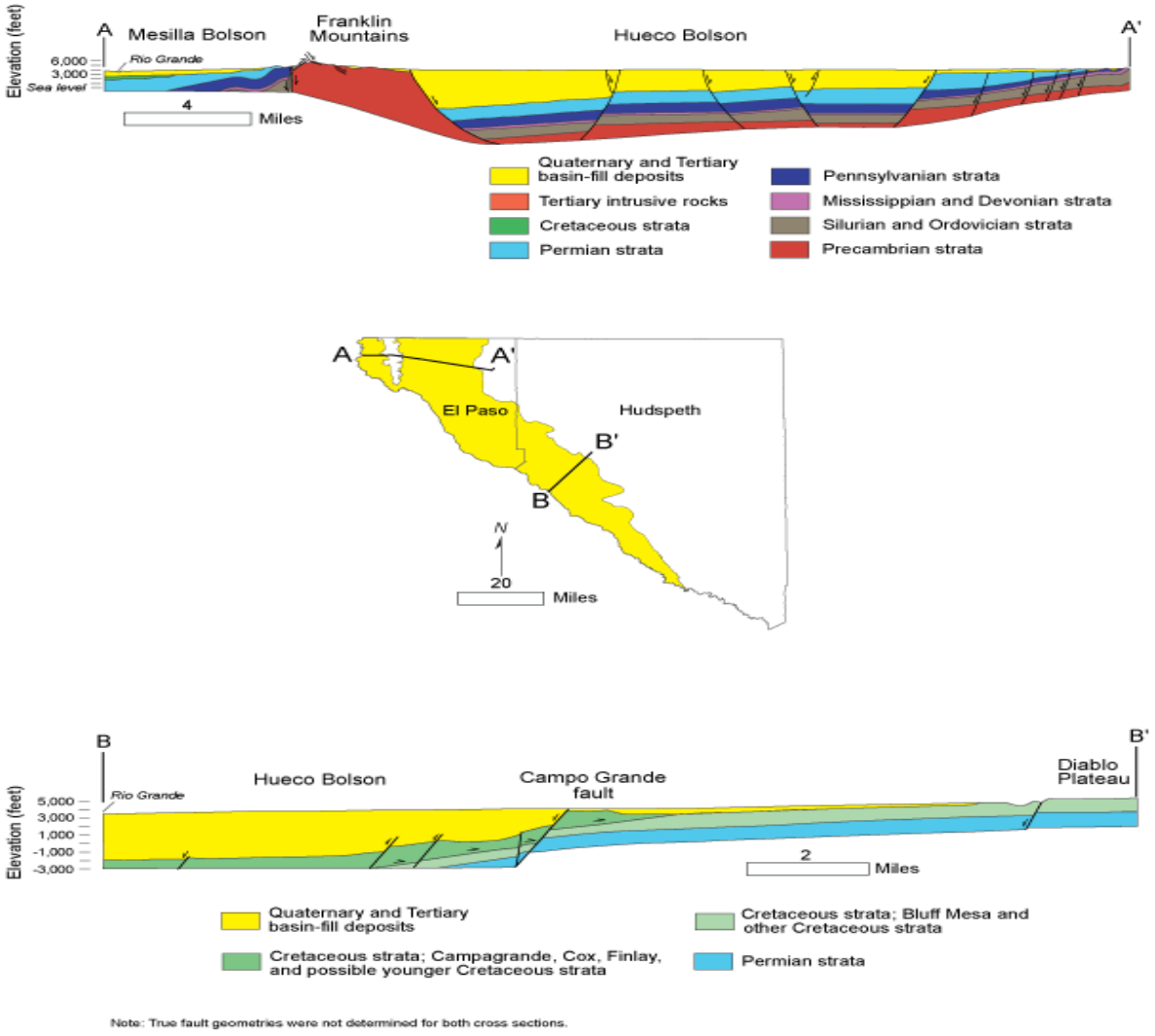


Figure 4:

This figure illustrates the different ages of strata that lie beneath the Hueco Bolson. Modified from Collins and Raney (1991).

Section 4 – The Santa Fe Group

The Hueco Bolson is up to 3000 m (9,000 ft.) thick and consists of deposits of gravel, sand, silt and clay (Mattic 1967; Gates et al. 1980). These deposits are termed the "Santa Fe Group", a complex sequence of basin-filling sedimentary rocks and some associated volcanic rocks. The Santa Fe Group ranges in age from middle Miocene to Pleistocene (Baldwin 1956). The Santa Fe

Group has been subdivided into four main units consisting of the Upper Santa Fe, Middle Santa Fe, Middle Lower Santa Fe, and Lower Santa Fe (Hawley et al. 2009).

Section 5 – The Aquifer

The upper portion of the Hueco Bolson contains fresh to slightly saline water, ranging from less than 1,000 (considered potable water) to 3,000 milligrams per liter of total dissolved solids. Its salinity typically increases to the south and in the shallower parts of the aquifer. Water level declines have contributed to higher salinity. Water is essential to the social, economic, and environmental well-being of Texas, especially in the arid areas of the western part of the state, where water is scarce and highly valued. Drought and increasing demand, primarily because of a growing urban population, have heightened concerns over water resources in the area.

Section 6 – Volume Reconstruction

The goal of this study is to generate a volumetric model of the basin using the interpreted data (Figure 5) from Hawley et al. (2009). However, upon evaluating the data it became clear that the current methods of volume reconstruction would not be applicable here. Cross-section H-H' illustrates geologic units that are not found in the adjacent cross-sections (G-D', and I-I') (Figure 5). This ruled out the use of surface based modeling due to the fact that I would not be able to generate surfaces for the extra units found in cross-section H-H' (Figure 5).

Attention then shifted to volume based modeling techniques, but the lack of a structural framework (i.e., how the faults are interconnected between the different cross-sections) ruled out the most popular method which used the interpolation of relative geological age across an unstructured grid (Figure 5). I decided to solve the problem as a classification problem and experiment with different machine learning algorithms to attempt to reconstruct a volumetric

model using the interpreted data (Figure 5) from Hawley et al. (2009) as the training dataset, and predicting across the entire grid space. Three algorithms were tested, Support Vector Classification (SVC), Gaussian Process Classification (GPC), and Probabilistic Neural Network Classification (PNNC).

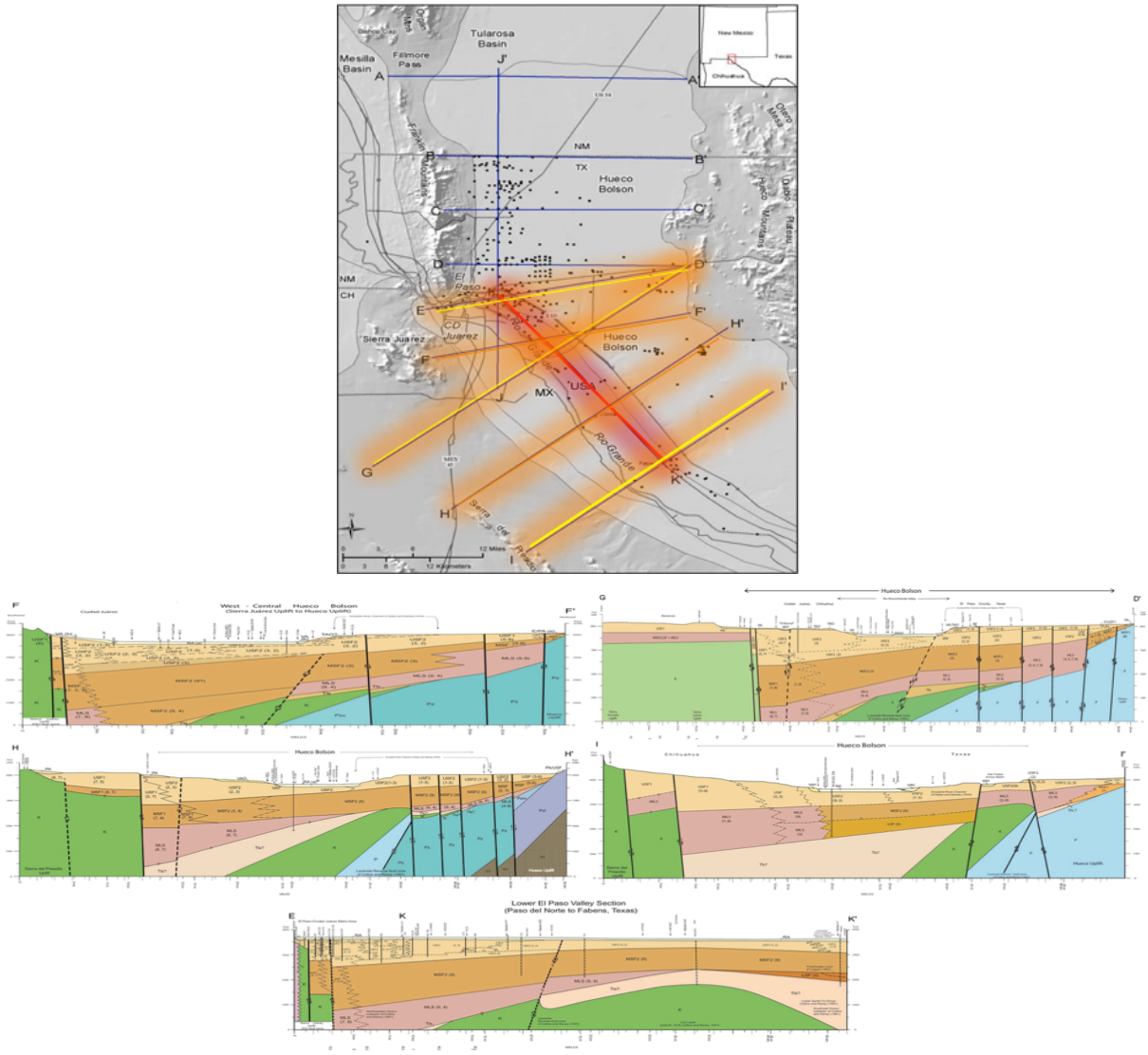


Figure 5:

Cross-section index map (top), showing location of schematic hydrogeologic sections (Hawley et al. 2009), and the five sections that were used in my preliminary study (bottom).

Chapter 2: Support Vector Machines

Section 1 - SVM for Volume Reconstruction

The use of Support Vector Machines (SVM) for geologic volume reconstruction has been studied by Smirnoff et al. (2008). Smirnoff et al. used 11 arbitrary chosen parallel sections from a 3D geologic model and attempted to reconstruct the 3D model using the SVM algorithm with an RBF (Radial Basis Function) kernel. Up to a 97.79% overall success of prediction was achieved with the use of SVM (Smirnoff et al. 2008). In my implementation of the SVM algorithm, I addressed the issue of success of reconstruction for a particular class being directly proportional to the number of those class points in the training set (Smirnoff et al. 2008) by applying the methodology behind Monte Carlo Integration, a technique used for numerical integration using random numbers. This technique is discussed in later chapters.

Section 2 - Overview

The SVM algorithm has its origins in The Nature of Statistical Learning Theory developed by Vapnik (1995). SVM algorithms works by classifying distinguishable data points and drawing a hyperplane or polyline that can separate the two data classes without error. SVM algorithms differ from other machine learning algorithms due to the fact that they place a larger emphasize on the data points that are most difficult to distinguish between two classes (Figure 6). The motivation behind the SVM algorithm is that if a classifier can succeed in distinguishing the most challenging comparisons (points in A and B that are closest to each other, also known as Support Vectors) (Figure 6), then such a classifier will be able to handle easier comparisons with greater success (points in A and B that are further away from each other) (Figure 6).

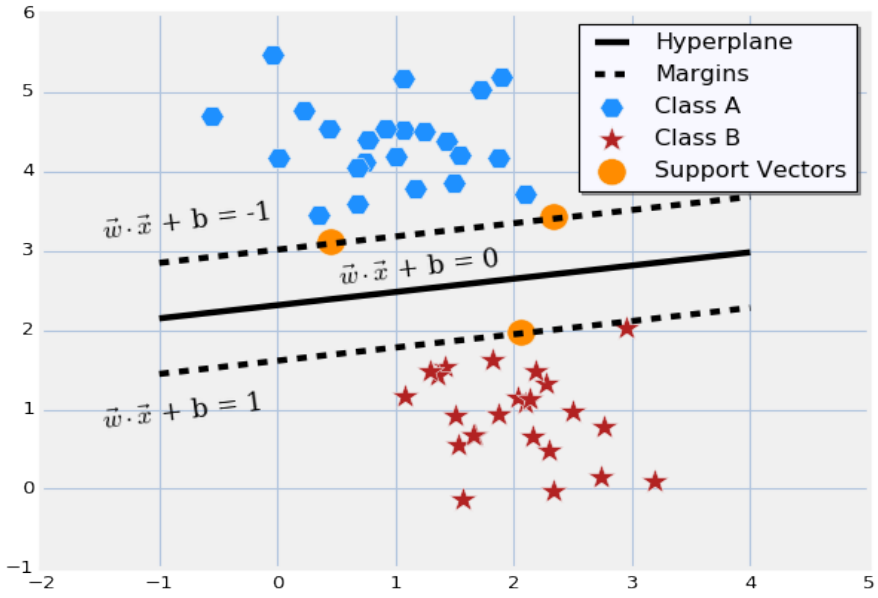


Figure 6:

An optimal hyperplane is drawn to distinguish between two classes (A/B) using the SVM algorithm. The dotted lines represent the "margins" for the two classes. The margins are determined by the support vectors. The optimal hyperplane lies halfway between these two margins.

Section 3 - The Optimal Hyperplane

The two margins in Figure 6 can be used to define the two classes as:

$$w \cdot x_i + b \geq -1, \forall x_i \text{ of Class A} \quad (1)$$

$$w \cdot x_i + b \leq +1, \forall x_i \text{ of Class B} \quad (2)$$

Where the vector w, also known as the width margin, is perpendicular to the hyperplane.

The optimal hyperplane can be defined by:

$$w \cdot x + b = 0 \quad (3)$$

To find the optimal hyperplane we need to maximize $\frac{2}{||w||}$, which is the perpendicular distance between the two margins (Figure 6) while meeting the constraints in equations 1 and 2. The aim of SVM is to orient the hyperplane in such a way that it is at the maximum distance from the support vectors of both classes. This requires us to find the variables *w* and *b* by solving the following objective function using Quadratic Programming:

$$\min \frac{1}{2} \left|\left|w\right|\right|^2$$

$$s.t.\ y_i(w \cdot x_i + b) \geq 1,\ \forall x_i \quad (4)$$

Perfect separation may not always be possible; in these situations, SVM finds the hyperplane that maximizes the margin and minimizes misclassification. This is where a *slack variable* g_i, is introduced to allow some instances to fall off the margin but penalize them. A regularization parameter, often termed as the *C* parameter, is introduced to control the degree of misclassification of each training example. Our new objective function takes the form:

$$\min \frac{1}{2} \left|\left|w\right|\right|^2 + C \sum_i^n g_i$$

$$s.t.\ y_i(w \cdot x_i + b) \geq 1 - g_i,\ \forall x_i,\ g_i \geq 0 \quad (5)$$

Section 4 - Nonlinearly Separable Data

A problem arises when our data are not linearly separable (Figure 7A). SVM deals with this issue by implicitly mapping the training data to a higher-dimensional space, commonly known as feature space $\phi(x)$ (Figure 7B). This transformation is helpful in converting nonlinear relations into linear relations (Cover's theorem). This is accomplished through the use of kernel methods;

a kernel is a dot product in some feature space. Some popular kernels for SVM include: Linear kernel, Gaussian kernel (Rbf), Exponential kernel, Polynomial kernel, Hybrid kernel, and Sigmoidal kernel (Guyon et al. 1993) (refer to Equations 6 to 9).

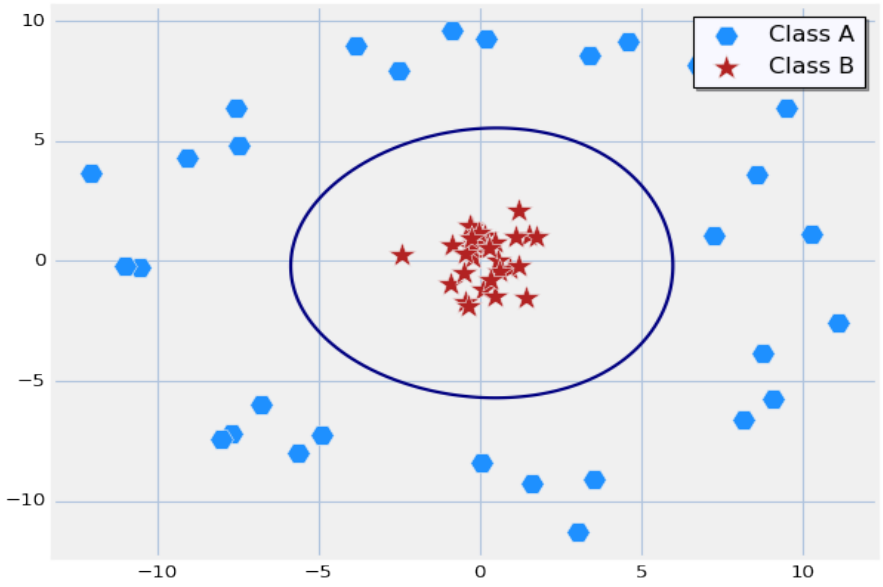


Figure 7A:

An example of data that is nonlinearly separable by a 2D hyperplane.

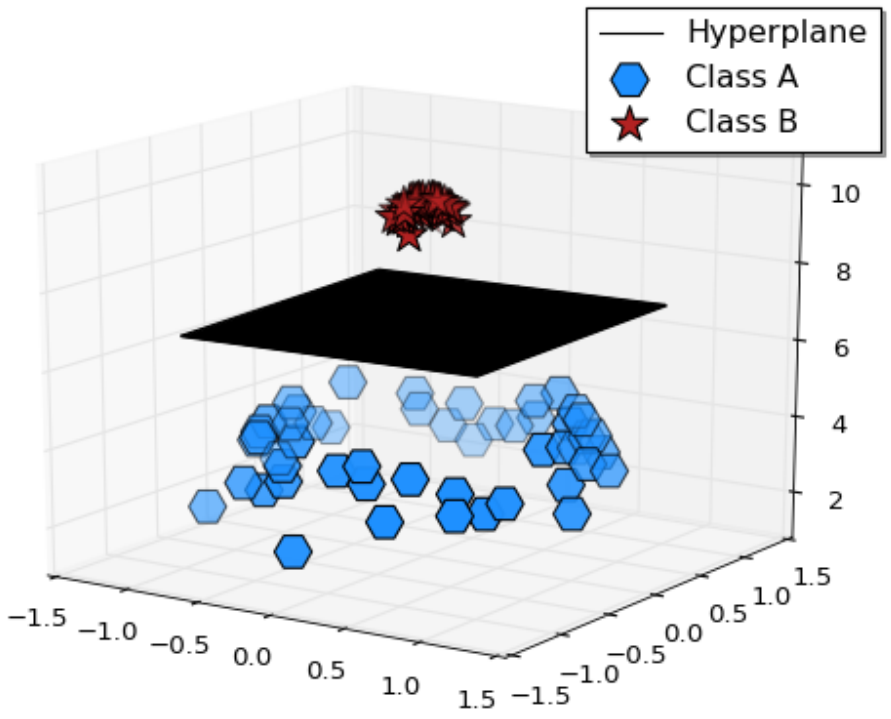


Figure 7B:

Here we use an exponential kernel to map the test dataset in Figure 7A to a 3D feature space where a linear hyperplane can be drawn. Once the optimal hyperplane is computed, it is mapped back to the original input space. The result is the hyperplane (solid line) in Figure 7A.

$$K(x, x') = (x \cdot x') \quad \text{(Linear Kernel)} \quad (6),$$

$$K(x, x') = (\gamma(x \cdot x') + r)^d \quad \text{(Polynomial Kernel of order } d\text{)} \quad (7),$$

$$K(x, x') = exp(-\gamma \|x - x'\|^2) \quad \text{(Rbf Kernel; } \gamma \text{ (gamma) > 0)} \quad (8),$$

$$K(x, x') = \tanh(\gamma(x \cdot x') + r) \quad \text{(Sigmoid Kernel)} \quad (9),$$

Chapter 3: Artificial Neural Networks

Section 1 - Overview

An artificial neural network (ANN) is an interconnected group of nodes which mimics the vast connection of neurons in a brain. Each circular node represents an artificial neuron, and each arrow represents a connection between neurons, known as edges (Figure 8). Edges have a weight associated with them, which determines the strength of a connection between two neurons. Data move from the input nodes and then get multiplied by their respective weights as they progress to the next layer of neurons (i.e. a Feedforward Neural Network). Each neuron in the ANN (with the exception of the input neurons) usually contains an activation function, which simply put, calculates a weighted sum of its inputs, adds a bias, and decides whether it should be fired (i.e. proceed to the next layer of neurons). Some of the most popular activation functions seen in ANNs include; Sigmoid or Logistic, Hyperbolic Tangent (Tanh), Rectified Linear Units (ReLu), and Statistically derived functions (Gaussian) in the case of Probabilistic Neural Networks (PNN). An ANN can be used for unsupervised learning, supervised learning, and reinforcement learning. ANNs can learn any function by readjusting the weights in the network in order to minimize a cost function using a combination of *gradient descent*, and *back propagation;* a technique which provides a computationally efficient method for evaluating derivatives in the network.

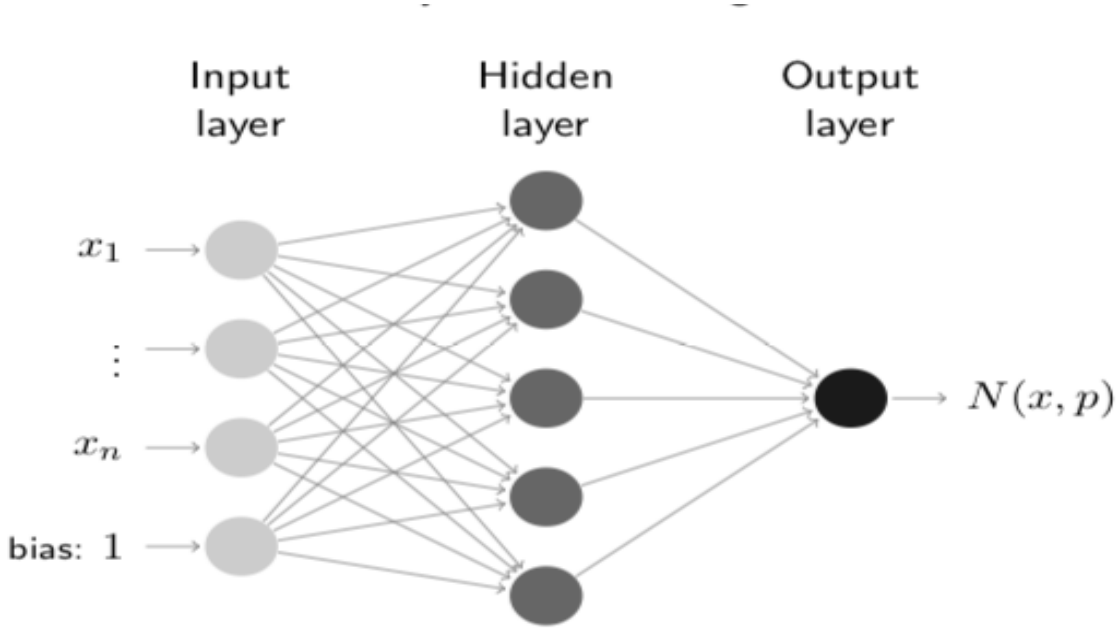


Figure 8:

A feed forward artificial neural network with 3 layers; an input layer with n + 1 nodes, a hidden layer with H nodes, and an output layer with 1 node. The arrows represent connections between the nodes (Chiaramonte and Kiener 2013).

Section 2 - Supervised Learning

For supervised learning (classification and regression), the ANN is first trained using a training dataset whose inputs and outputs are already known. The goal is for the ANN to consume the inputs from the training dataset and produce the appropriate outputs. The initial weights of the ANN are set at random and so ANN will usually produce incorrect outputs on the first run. This is where the ANN modifies the weights in order to produce the correct outputs by first using *back propagation* to compute the gradients, then apply *gradient descent* to compute improved weights and minimize the objective function. Some popular objective functions for classification include; square loss, hinge loss, logistic loss, and cross entropy log loss (Equation 10). Some popular objective functions for regression problems include; mean absolute error, mean absolute percentage error, squared hinge, and mean squared error (Equation 11).

The cross entropy log loss between the ground truth (true output of the training dataset) p and the ANN's output q, used in classification problems is:

$$CE = -\sum_x p(x) \log q(x) \quad (10)$$

The mean squared error between the ground truth (true output of the training dataset) y and the ANN's output $\hat{y}$, used in regression problems is:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (11)$$

Section 3 - Adam

There are 3 flavors of gradient descent (batch gradient descent, stochastic gradient descent, and mini-batch gradient descent), which differ in how much data they use to compute the gradient of the objective function. Adaptive Moment Estimation (Adam) is a mini batch variant of gradient descent. This is a preferred method of training a neural network and optimization in general due to it computing adaptive learning rates for each parameter. An exponentially decaying average of past squared gradients $v_t$and an exponentially decaying average of gradients $m_t$are stored by Adam (Kingma et al. 2015) (equations 12 and 13).

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\frac{\partial J}{\partial \theta_t} \quad (12)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\frac{\partial J}{\partial \theta})_t^2 \quad (13)$$

Kingma et al. (2015) observed that $m_t$and v_t are biased towards zero when the decay rates are small; β_1 , β_2 are close to 1. This was counteracted by computing the respective bias corrected estimates $\widehat{m_t}$, and $\widehat{v_t}$ (Kingma et al. 2015).

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \quad (14)$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t} \quad (15)$$

Kingma et al. (2015) proposed default values of 0.9 for β_1, 0.999 for β_2, and 10^{-8} for ε. The Adam update rule is defined as:

$$\theta_{t+1} = \theta_t - \frac{\propto \widehat{m_t}}{\sqrt{\hat{v}_t}+\varepsilon} \quad (16)$$

Section 4 - Probabilistic Neural Networks

Probabilistic Neural Networks (PNN) are a type of feed forward neural network used in classification and pattern recognition. This type of neural network contains 4 layers of nodes in its architecture (Specht 1990); an *input layer* which contains *N* nodes, one for each input feature with set measurements representing a predictor variable, a *pattern layer* which contains *K* classes consisting of the Gaussian functions formed using the given training dataset, a *summation layer* which performs a scaled sum for the Gaussian values for the *K* classes to form a probability density function, and an *output layer* which selects the largest value determining the associated class label (Figure 9). Unlike other ANNs, the weights in a PNN do not have to be modified.

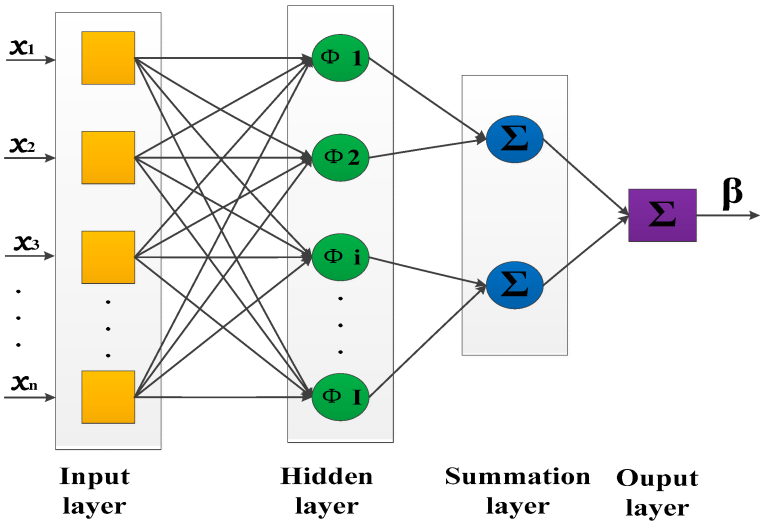


Figure 9:

Structure of a PNN illustrating input layer, pattern layer also known as hidden layer, summation layer, and output layer (Xu et al. 2016).

PNN classifiers are inherently nonparametric and based on a Parzen window estimate of the joint distribution (i.e. estimate the probability density function of a random variable) (kernel density estimation). The Parzen window density estimation is computed by:

$$p(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h^2} \emptyset \left(\frac{x_i - x}{h} \right) \quad (17)$$

Where n is the number of elements in the vector, x is a vector, $p(x)$ is the probability density of x, h is the dimension of the Parzen Window, and $\phi$is a window function.

Chapter 4: Results

Section 1 - Generating Point Cloud Data

A custom digitizer (Figure 10) was developed with the Python programming language, leveraging bindings to QT for the graphic user interface, Numpy for fast array manipulation, Shapely generating polygons, Scipy and PIL for image processing, and Pandas for data manipulation. The software takes cropped images from Figure 5 as input, then prompts the user to assign 3 coordinates (longitude, latitude, elevation) to reference the image in 3D space. The longitude and latitude inputs must be in decimal degrees, and elevations must be in meters. Once the image has been referenced, the user can select an integer to represent a geologic formation and begin drawing polygons.

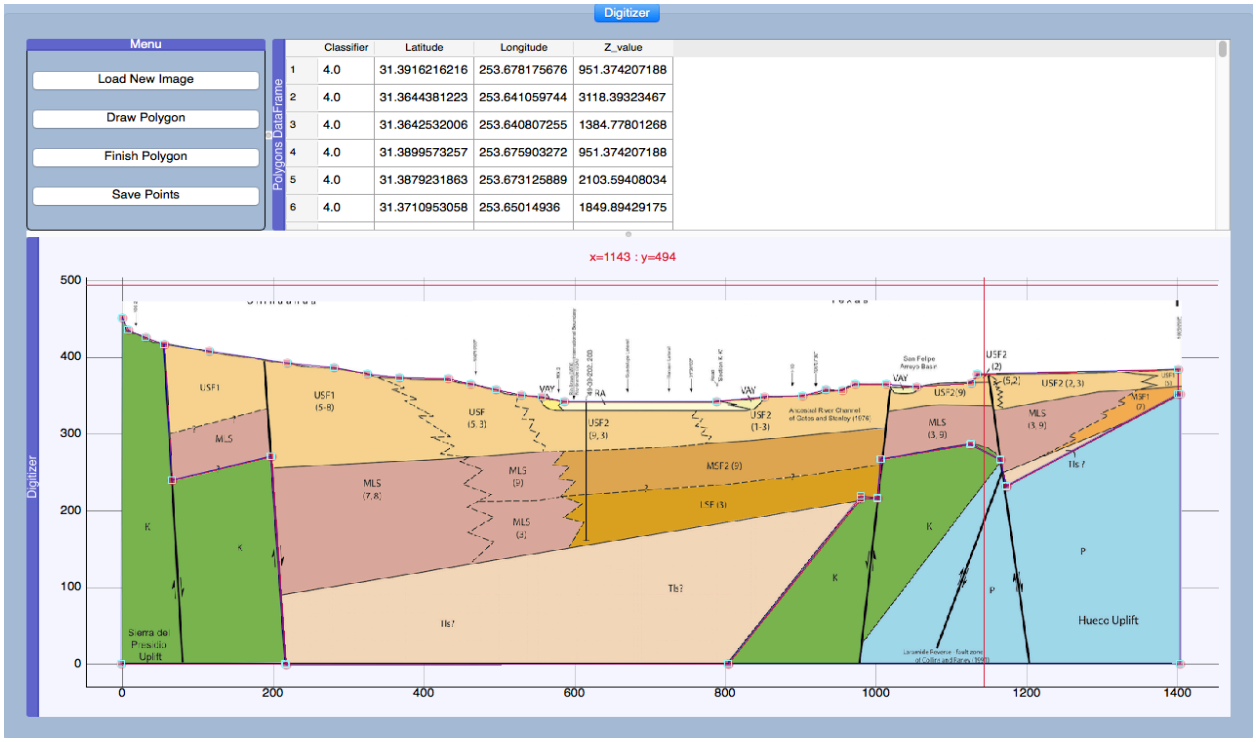


Figure 10:

Polygons being drawn over a cropped image of a geologic cross section (bottom), control buttons (top left), and table of point cloud data that has already been generated (top right).

Smirnoff et al. noticed the success of reconstruction for a particular class is directly proportional to the number of those class points in the training set with the SVM algorithm (Smirnoff et al. 2008). A solution to this problem was inspired by *Monte Carlo Integration.* For each polygon that is digitized, the software will generate 10,000 random points within the relevant polygon (Figure 11). Each data point contains a coordinate (longitude, latitude, elevation) and the assigned integer used to represent the geologic formation. This newly generated data will later serve as the training dataset for SVM, and PNN classifiers.

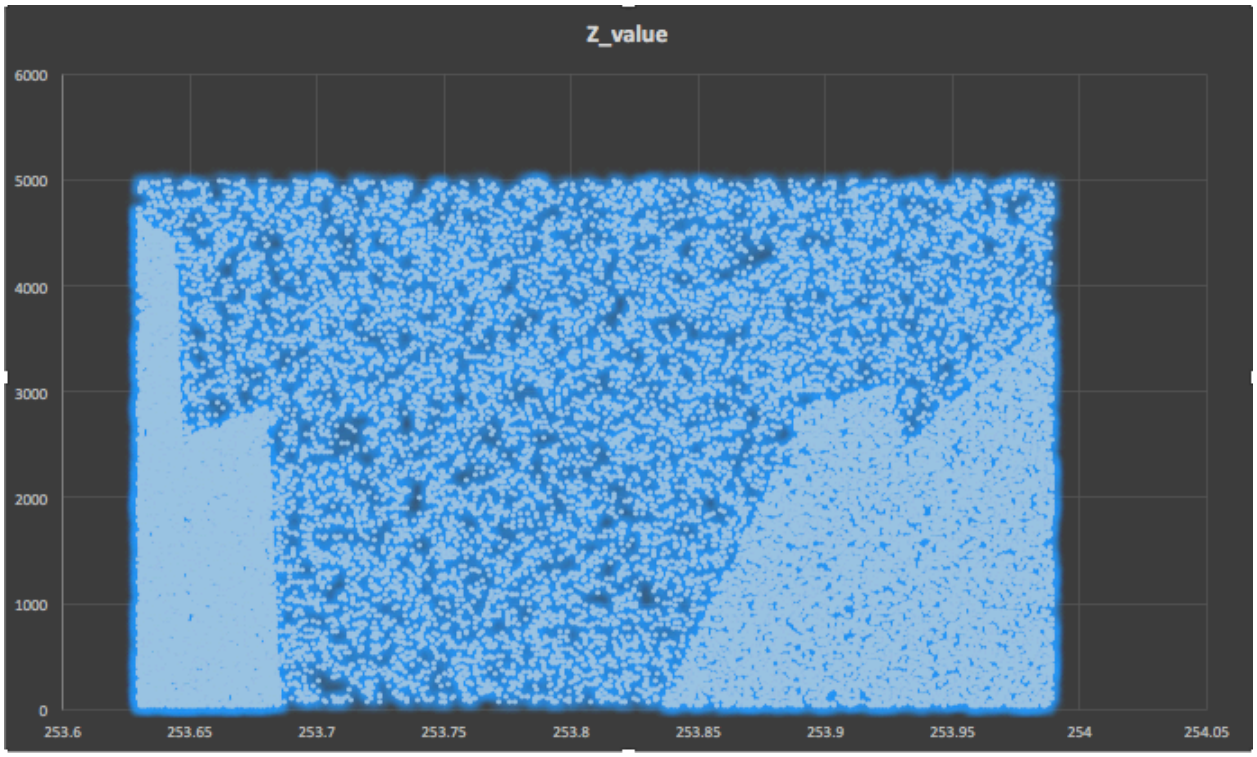


Figure 11:

Point cloud data generated by the custom built digitizer in Figure 10 that was used as input for SVM and PNN to produce a 3D model.

Section 2 - SVM

The training data were fed to the SVM classifier from Scikit-learn (Pedregosa et al. 2011) ; using an RBF kernel with gamma = 0.03 (Figure 4), and C=1 (Figure 12). The relatively high

gamma value restricted the SVM from predicting (i.e. extrapolating) too far away from the training dataset. This proved to be an essential feature to have, especially when dealing with sparse data. A geoscientist can effectively control the extent of extrapolation by controlling the value of gamma using an RBF kernel. Pyqtgraph, a pure Python graphics and GUI (Graphic User Interface) built on OpenGL bindings, was the preferred tool for rendering the volumetric models. The volumetric models are fully interactive (panning, zooming, rotating).

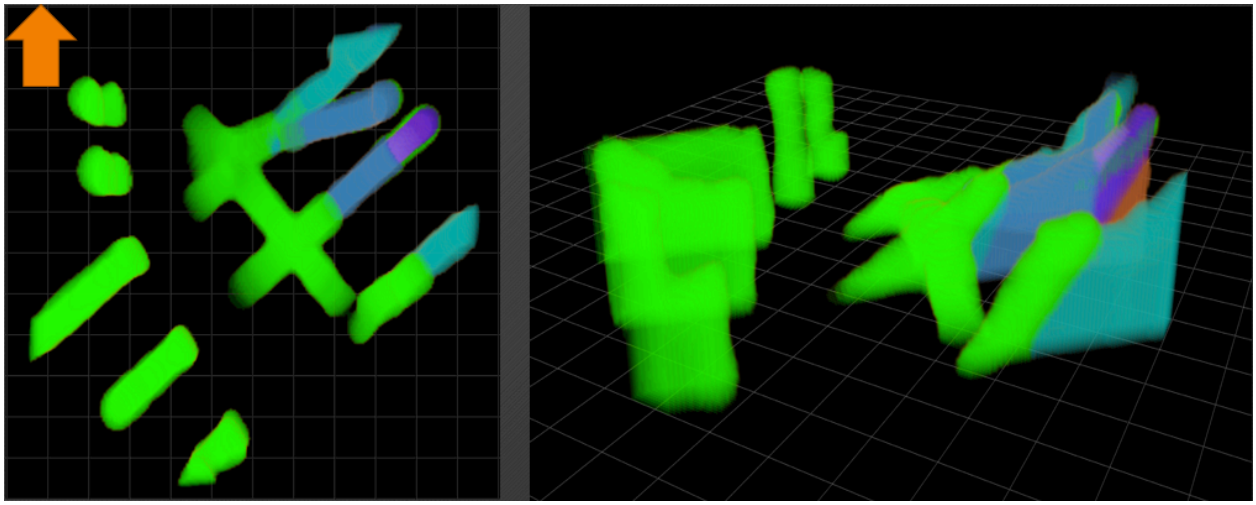


Figure 12:

Results from a SVM using an RBF kernel, C=1, gamma=0.03. Showing a reconstruction of the bedrock formation in the cross section data from Hawley et al. (2009) (Figure 6) in 3D space. The dimensions of this model are 300x300x100 voxels.

A GUI was developed to improve usability, testing, ease of construction of volumetric models, and interaction with the models (Figure 13). A new volumetric model was constructed with the software using the SVM algorithm with an RBF kernel, gamma = 0.0003, and C=1 (Figure 13). The reduced value of gamma gave the SVM the freedom to predict (extrapolate) across the entire model space. The software allows the geoscientists to slice the model horizontally (bottom right), and slice the model vertically (top right) generating synthetic cross

sections of the volumetric model. These features allow the geoscientist to analyze the resulting volumetric models.

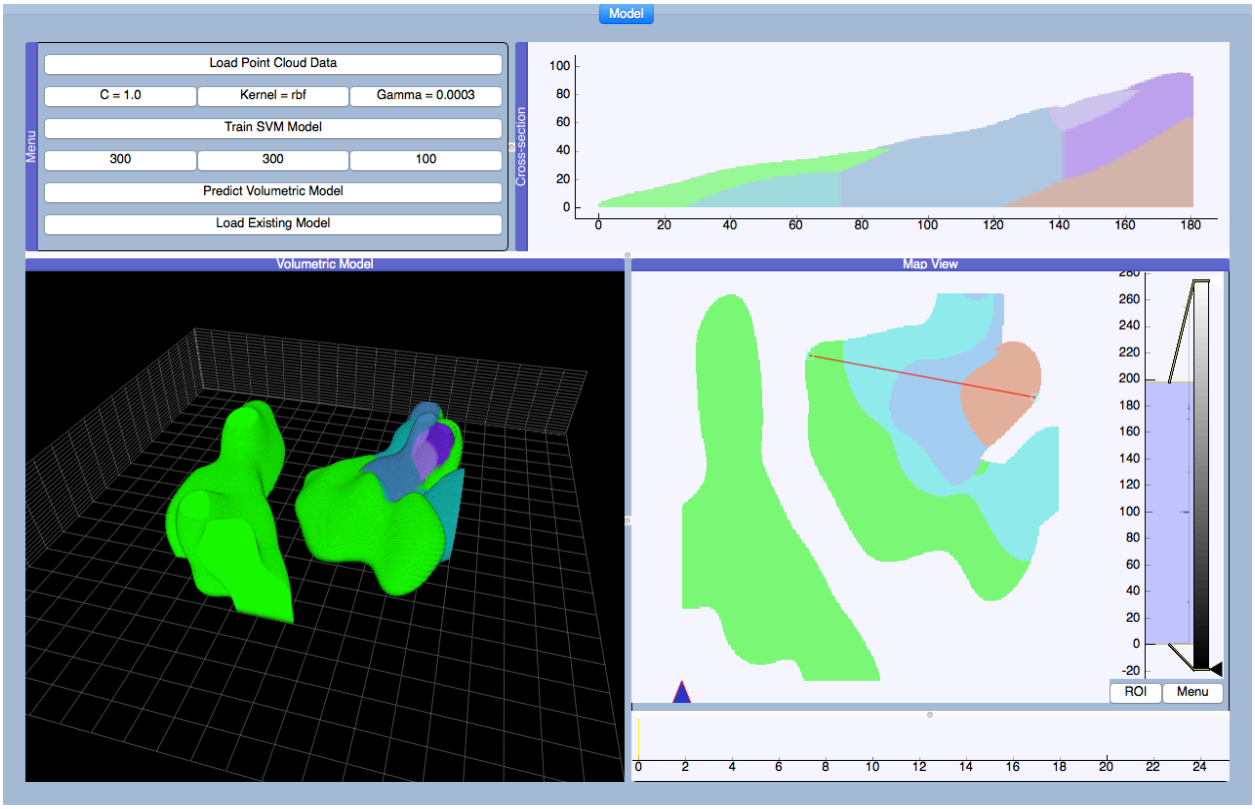


Figure 13:

Results from a SVM using the RBF kernel, gamma=0.0003, and C=1 for the same point cloud data used to generate the model in Figure 12. The volumetric model is shown (bottom left), a horizontal slice of the model (bottom right), and a synthetic cross section of a portion of the model indicated by the red line (bottom right) is illustrated in the top left of the GUI. The dimensions of this model are identical to those found in Figure 12.

The SVM classifier from Scikit-learn is very mature; granting the user access to rich details from the resulting models. The mean accuracy of the training dataset resulting from the SVM model was 0.97127. There are 6 bedrock units being modeled; a Precambrian igneous unit (XY), Lower Paleozoic Unit (Pzl), a Middle Paleozoic (Pzm), a Paleozoic Unit (Pz), a Permian Unit (P), and a Cretaceous Unit (K) (Figure 5) (Hawley et al. 2009). The SVM algorithms used a total of 4,794 support vectors to generate the volumetric models (Table 1).

Table 1:

Number of Support Vectors for Each Class (Geologic Unit) in training dataset.

Geologic Unit (Class)	Number of Support Vectors
XY	307
Pzl	184
Pzm	187
Pz	722
P	1050
K	1244
Overburden	1100

Section 3 - PNN

The same training data were fed to a PNN algorithm from Neupy, a python library for neural networks (http://neupy.com/pages/home.html) with batch size = 128, step=0.1, and std=12 (Figure 14). The model produced by PNN closely resembles the SVM model in Figure 13. The general shape of the rock formations and main geologic structures were adequately captured by both algorithms. However, the PNN does not smooth the model to the same extent as a SVM, making it easier to identify certain traits such as possible strike-slip faults. These features could also be artifacts that stem from the orientations of the cross sections, and the scarcity of the data.

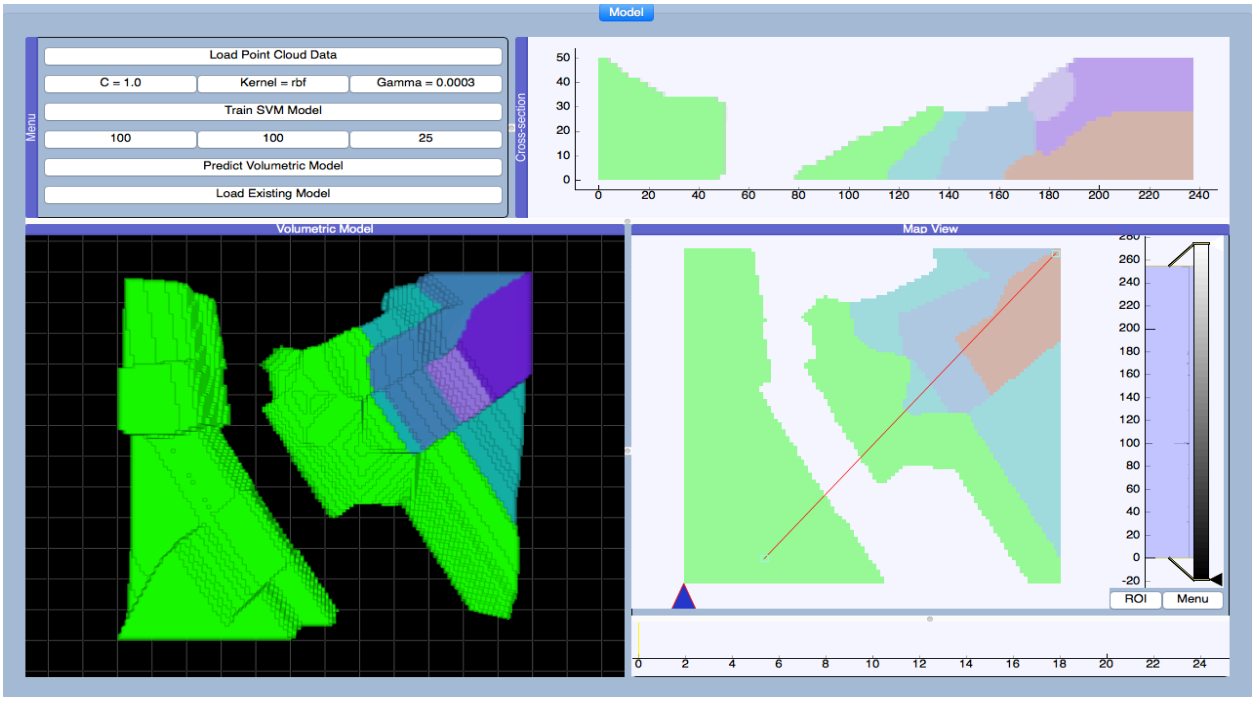


Figure 14:

Results from a PNN with batch size=128, step=0.1, std=12. PNN seems to outperform SVM when identifying possible strike slip faults. SVM produces results that are too smooth when dealing with sparse data.

Possible strike-slip faults were interpreted from the results of the PNN model (Figure 15). Most of the faults can be observed in the SVM model (Figure 13). However, they are not as prevalent and a non-experienced geoscientist could overlook them at first glance. It took a few seconds to train the PNN. However, classifying new data points and generating the complete volumetric models took approximately 7 minutes. The SVM algorithm took 40 seconds to train, and approximately 2 minutes to produce a volumetric model with similar dimensions (100x100x25 voxels).

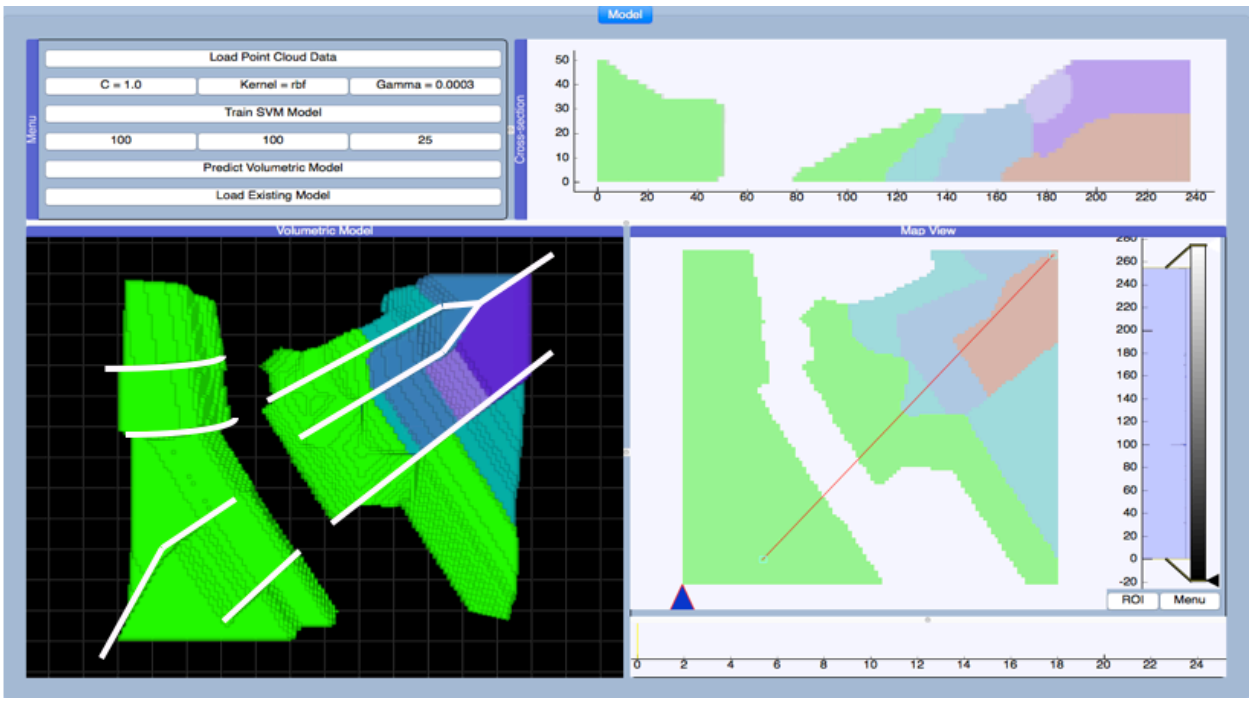


Figure 15:

Interpretation of possible strike slip faults from the volumetric model generated by the PNN algorithm. The volumetric model's dimensions are 100x100x25 voxels.

Section 4 - Comparison: SVM vs PNN

New volumetric models were produced with both algorithms (SVM, PNN) with identical dimensions (200x200x50 voxels) in order to compare the cross-sections from Hawley et al. (2009) to the corresponding cross-sections generated by SVM and PNN (Figures 16 – 21). Both algorithms performed well at this low resolution, although SVM clearly outperformed PNN in Figures 17, 18, and 21.

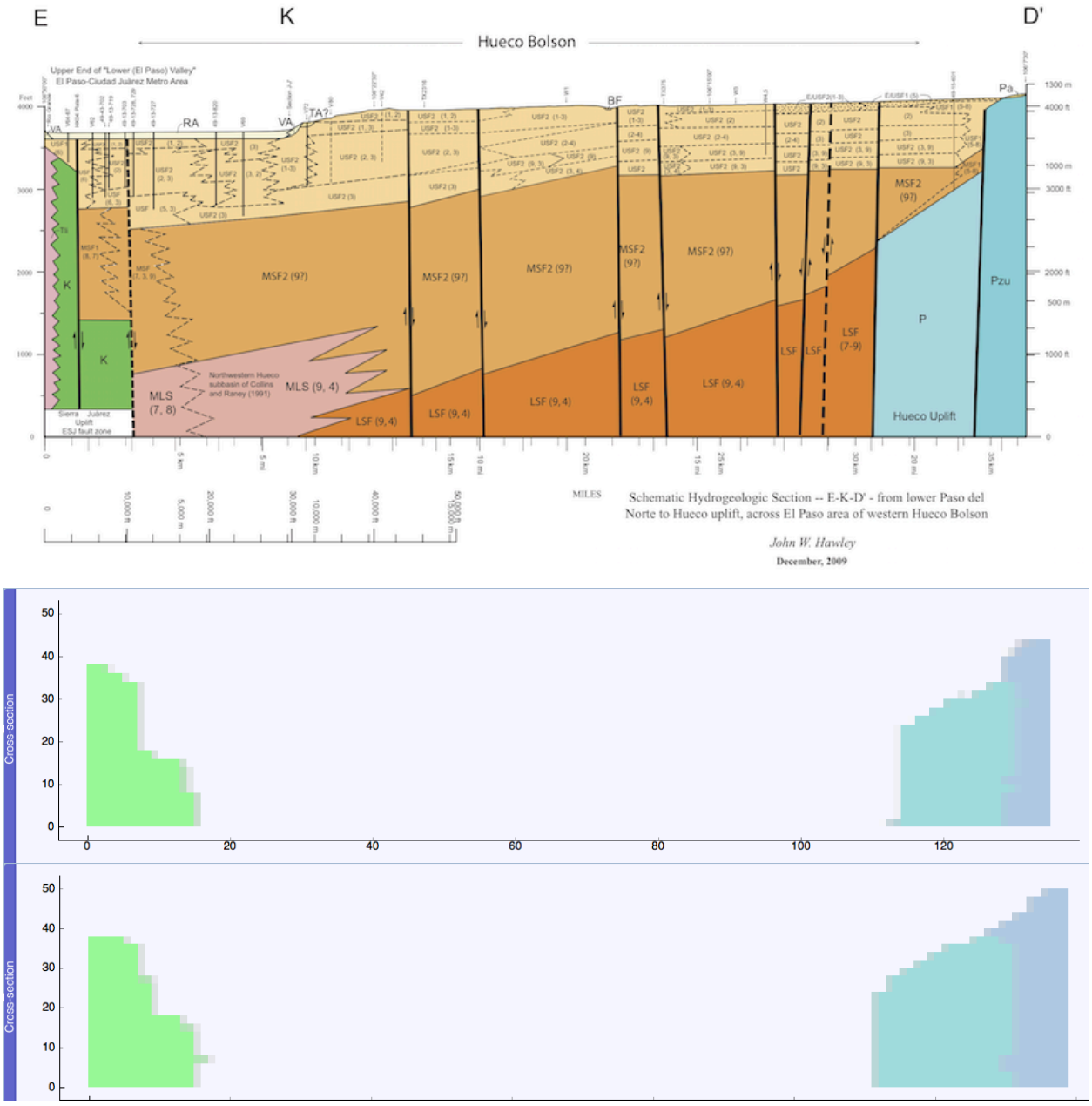


Figure 16:

Original cross section E – D′(top), cross sections reconstructed by SVM (middle), and PNN (bottom).

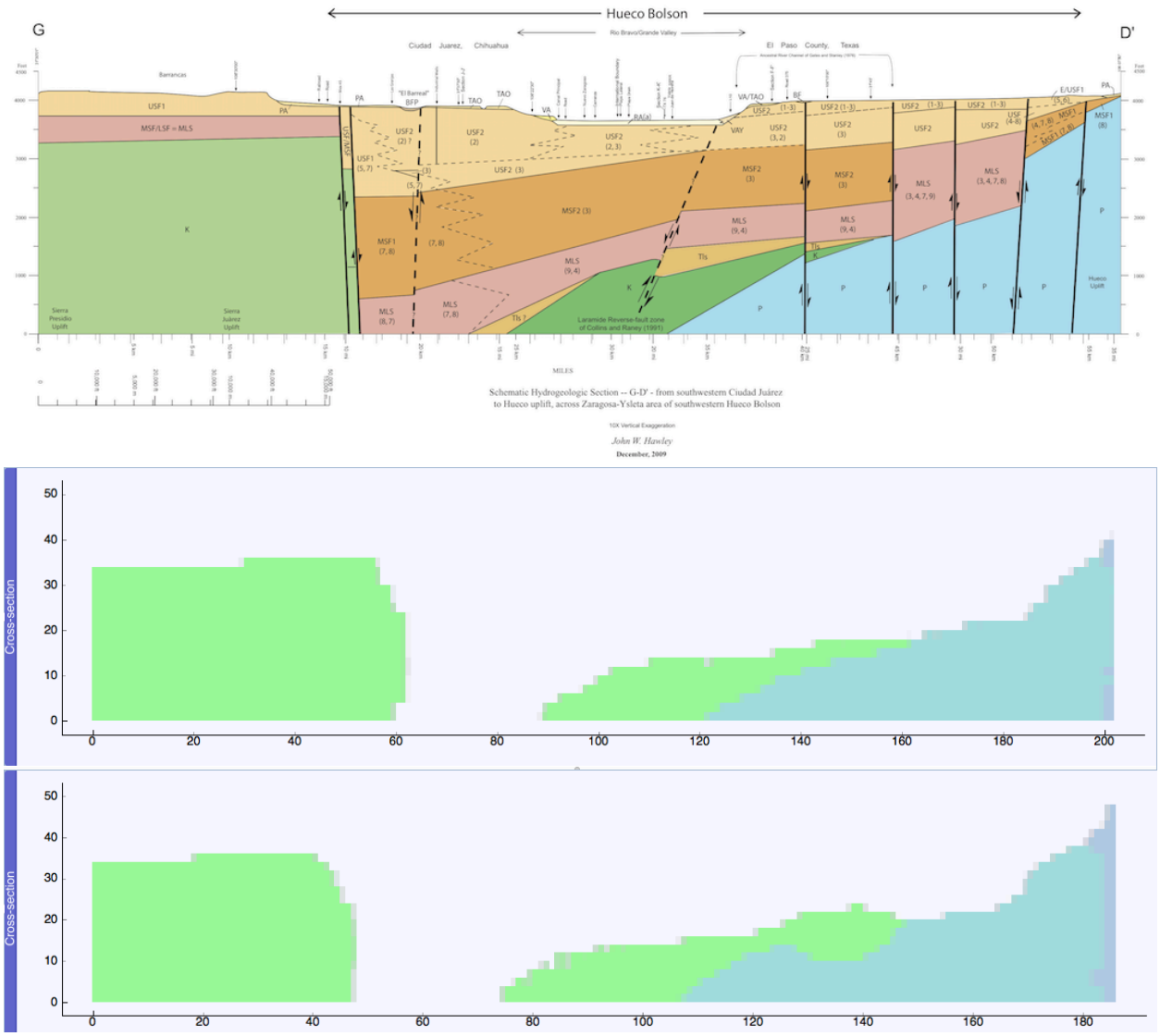


Figure 17:

Cross section G − D′(top), SVM (middle), PNN (bottom).

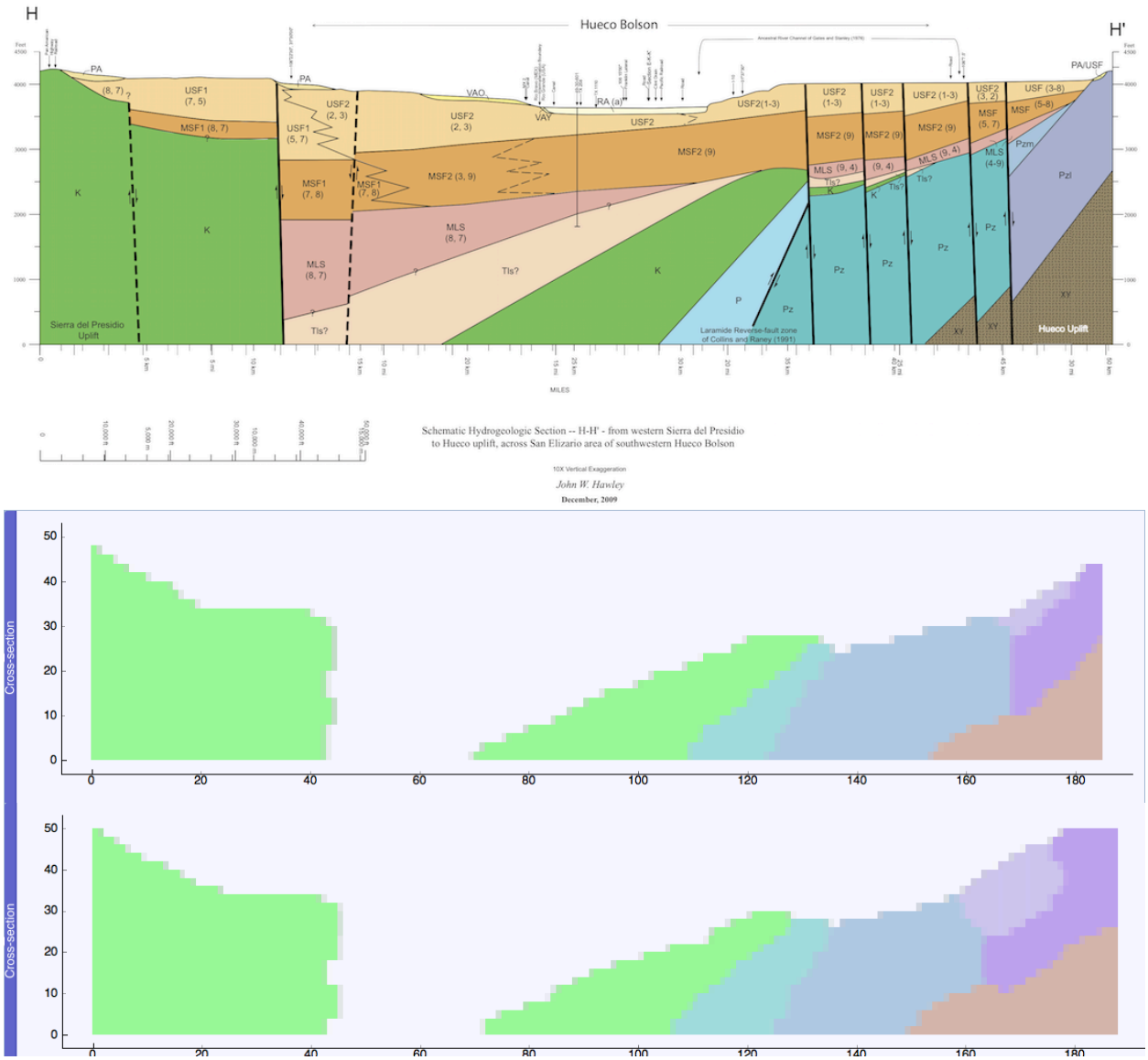


Figure 18:

Cross section H − H′(top), SVM (middle), PNN (bottom).

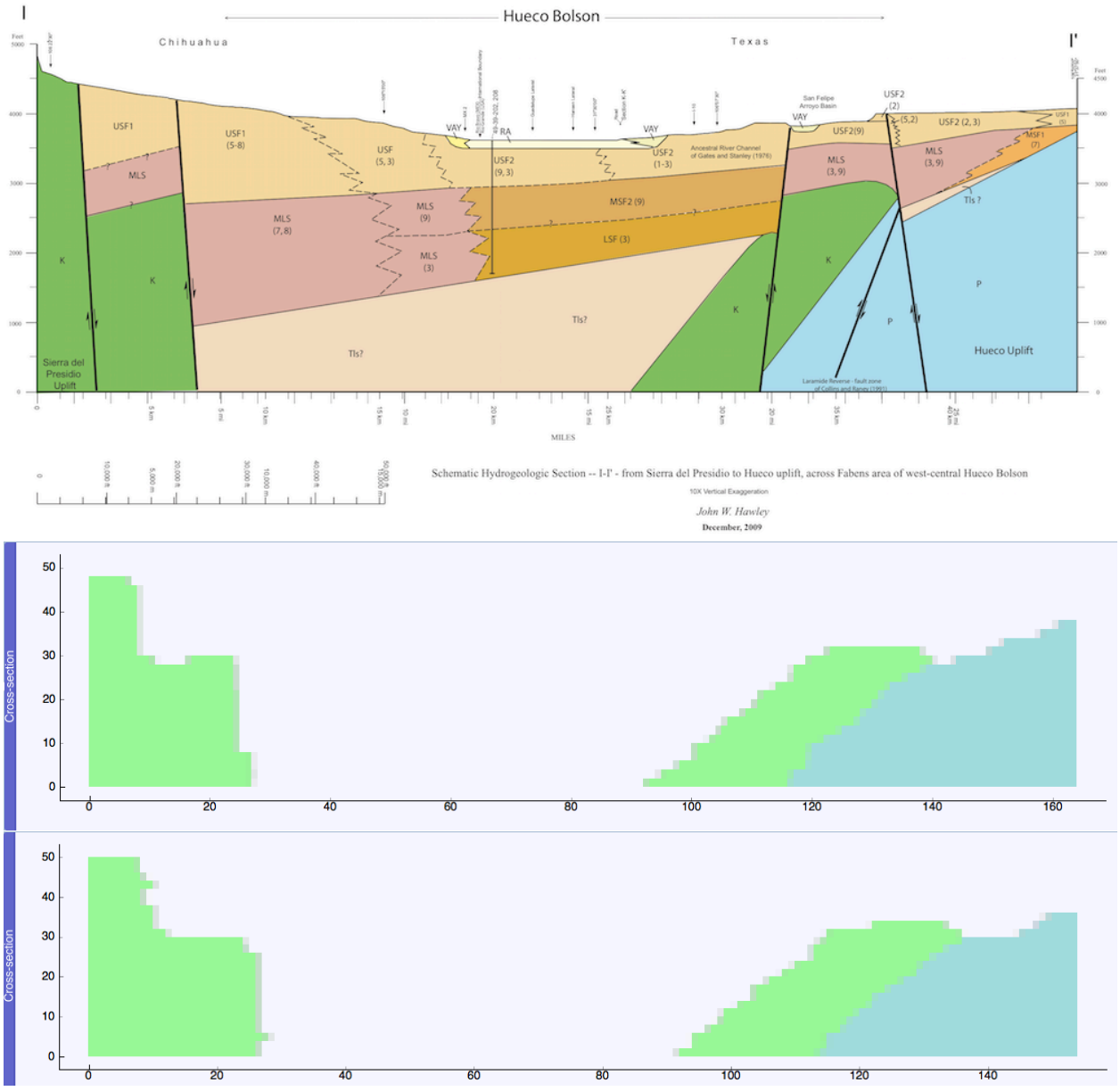


Figure 19:

Cross section I – I′(top), SVM (middle), PNN (bottom).

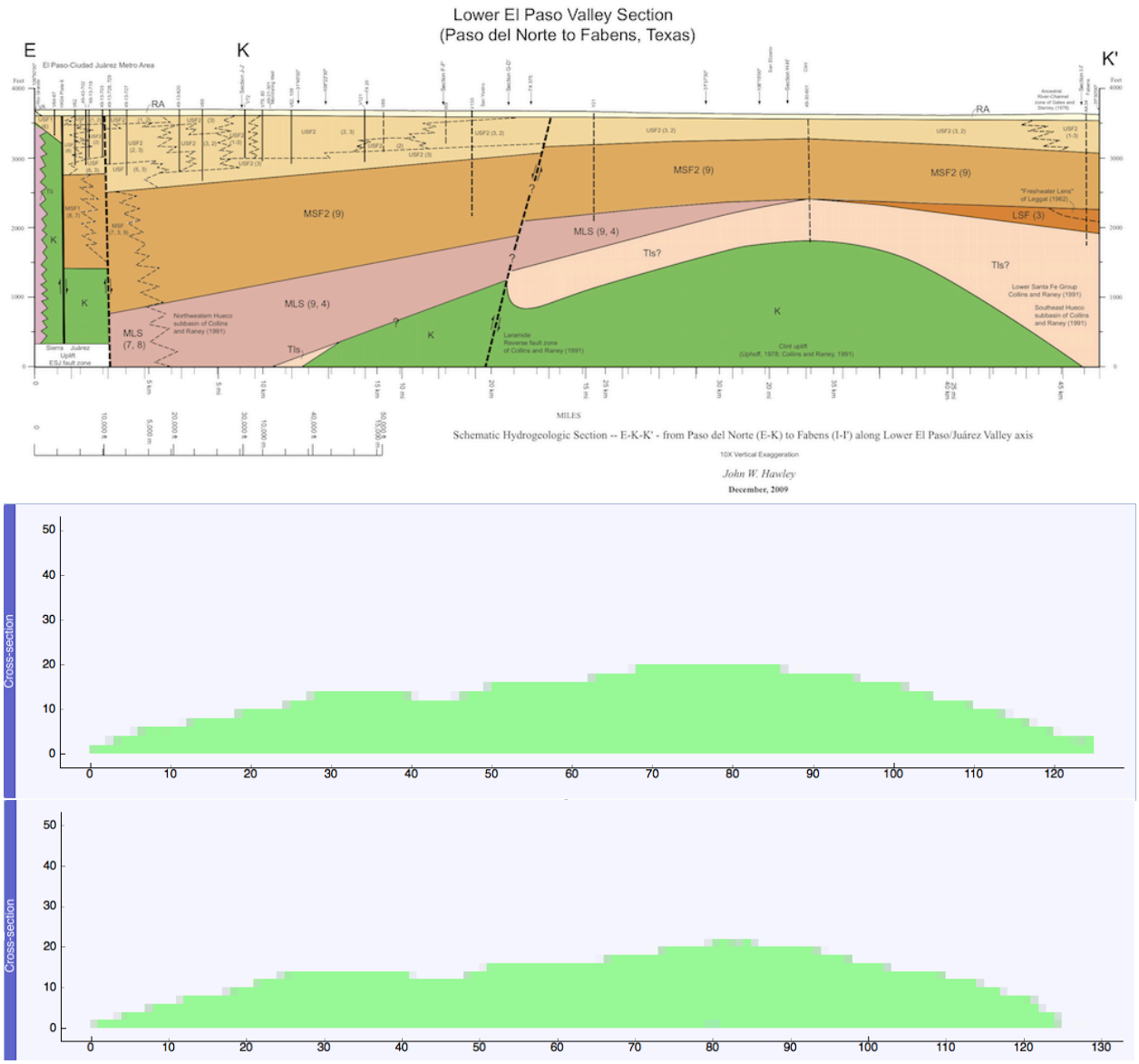
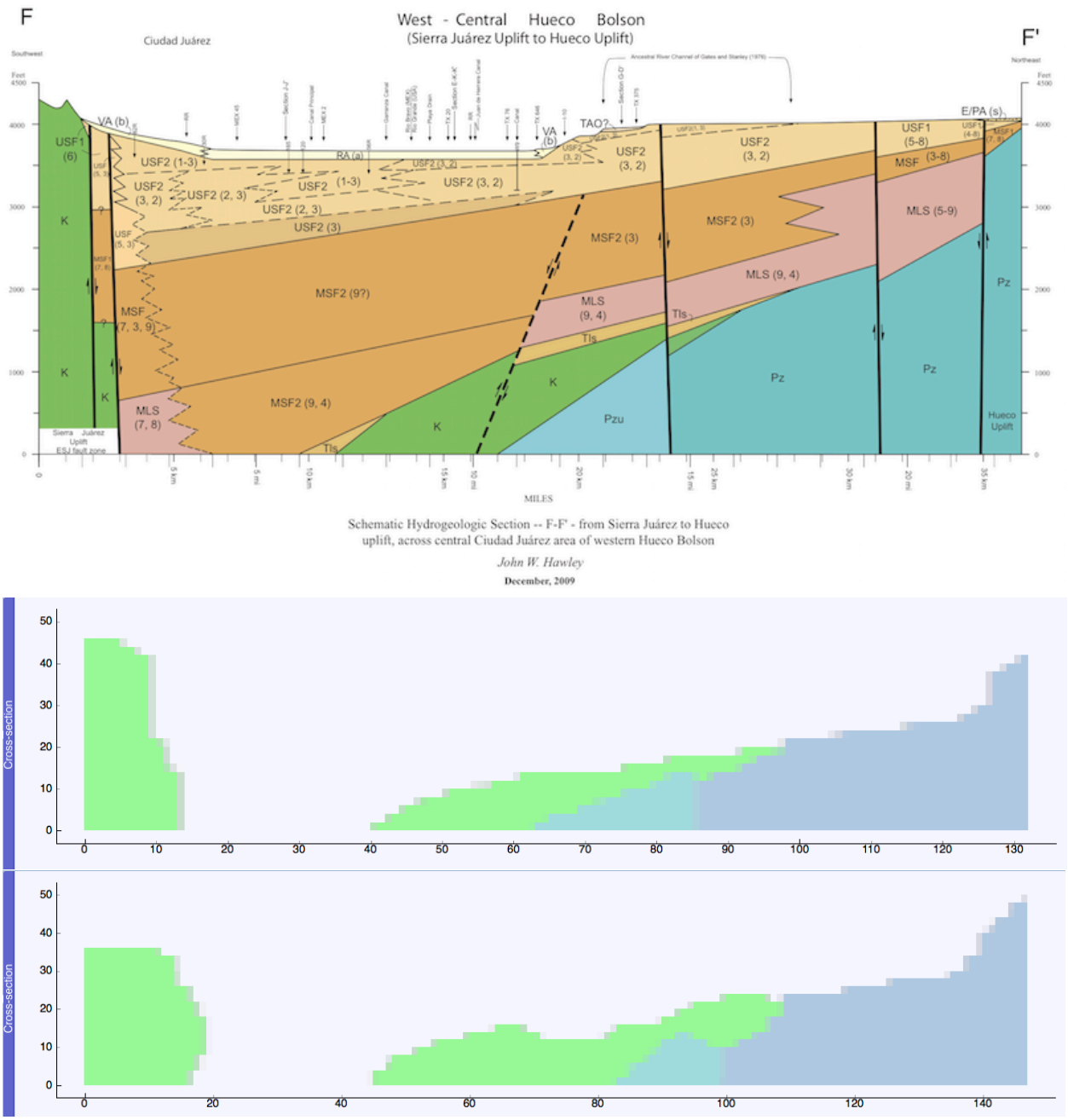


Figure 20:

Cross section E − K′(top), SVM (middle), PNN (bottom).



Schematic Hydrogeologic Section -- F-F' - from Sierra Juárez to Hueco uplift, across central Ciudad Juárez area of western Hueco Bolson

John W. Hawley

December, 2009

Figure 21:

Cross section F − F′(top), SVM (middle), PNN (bottom).

Chapter 5: Discussion

Section 1 - Interpretation of Volumetric Models

The strike slip faults predicted by the learning algorithms (PNN and SVM) can serve as an explanation for the discrepancy in the interpreted cross-sections (Figure 5) discussed in Chapter 1 Section 6. There are two historical geologic events that could have produced the strike slip faults in the area; the Laramide Orogeny, and the Rio Grande Rift.

The Laramide Orogeny was a period of mountain building that took place in western North America starting in the late Cretaceous (Figure 3). This coincides perfectly with the youngest bedrock unit in our models which is Cretaceous in age. The Laramide Orogeny is known to have produced strike slip faults in other areas in western North America. The strike slip faults in the Hueco Bolson could also have been reactivated during the Rio Grande Rift.

The Rio Grande Rift is a north trending continental rift zone which began forming in the early Miocene. It is responsible for the normal faulting in the Hueco Bolson, and could be responsible for the strike slip faults as well. Literature on the East African Rift suggests that rift zones are capable of producing strike slip faults under certain conditions (Abbate et al. 1995).

Section 2 - Representing Volumetric Models as Functions

Storing the volumetric models produced by these algorithms proved to be another challenge. In the age of cloud computing where data have to be moved from servers across the country or across the world, moving millions or billions of data points (voxels) can be time consuming. Take, for example, the volumetric model in Figure 13. This model contains 9 million voxels, each voxel contains 4 values (red, blue, green, and opacity) leading to a total of 36 million values to be stored on a spreadsheet or a database. However, this is an extremely low

resolution model; increasing the resolution to 1000x1000x500 voxels would yield 2 billion values.

A solution to this problem was achieved by representing a volumetric model as a function with the help of Deep Artificial Neural Networks. A 3 hidden layer neural network was developed (containing 100 neurons in the first hidden layer, 20 neurons in the second hidden layer, and 20 neurons in the third hidden layer) to approximate a function that could reconstruct the volumetric models to a degree of accuracy. The input for this neural network was the coordinate values of the entire model (x,y,z) and the matching output was the integer values assigned to each rock type in the model. The ANN was able to achieve a 95% accuracy of reconstruction (Figure 22) using sigmoid logistic activation functions, the Adam optimization algorithm (Equation 15), and a Cross Entropy objective function (Equation 10). The accuracy of the ANN can be improved by using a higher resolution model as input.

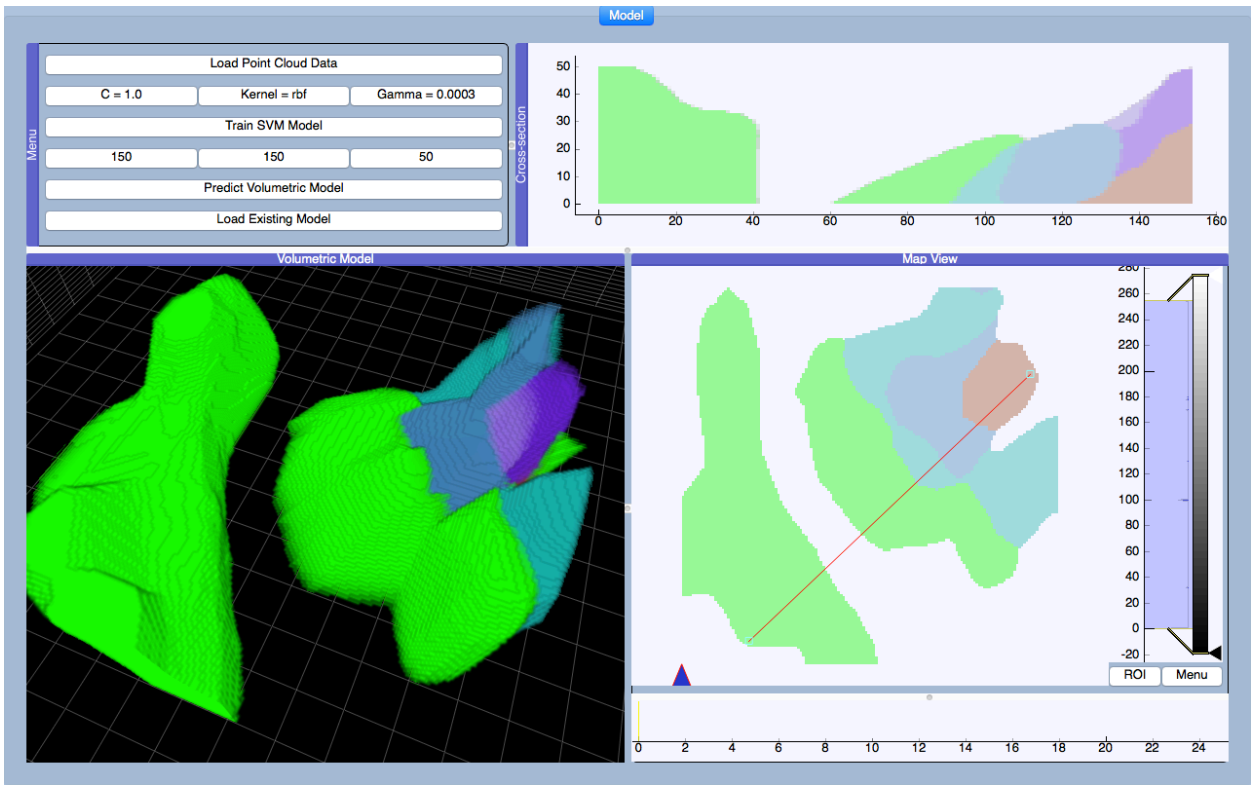


Figure 22:

The result of a deep artificial neural network's attempt to compress a volumetric model into a function. The input was a volumetric model generated with the SVM algorithm (150x150x50 voxels). The ANN succeeded in reconstructing the SVM model (compare to Figure 13).

The function produced by the ANN from Figure 22 resulted in 2,730 parameters (weights of edges in the ANN), which would have to be stored in a database or spreadsheet (compared to the 4,500,000 values of the original volumetric model). This demonstrates how ANN can be effective in reducing the storage space required by a volumetric model by compressing it into a function. It is also possible to modify the ANN from Figure 22 to solve a geophysical inverse problem, or use the function produced by the ANN as an input for methods such as finite differences, finite elements, or finite volumes.

Chapter 6: Conclusions

Section 1 - Assessment of Methods

The memory requirements for the GPC algorithm mentioned in chapter 1 section 6 proved to be too large for this application. When attempting to train the 30,000 points using the GPC algorithm, the memory usage maxed out at 50GB before the program was killed. Both SVM and PNN proved to be effective and applicable, but come with some tradeoffs; SVM was faster than PNN but was susceptible to smoothing, SVM with an RBF kernel granted more flexibility by allowing the user to decide how far to extrapolate from input data. SVM can produce artifacts when data are extremely sparse. PNN shows possible strike slip faults with more clarity but is more computationally intensive (generating large models could take days or possibly weeks).

In this study SVM and PNN complimented each other. The use of PNN is recommended to generate low resolution models when there is a need to map out geologic structures (e.g., possible strike slip faults) and when cross-sectional data are sparse. The use of SVM with an RBF kernel is recommended when generating large volumetric models, or when the geoscientist desires more control over the extent of extrapolation away from the input data. Both algorithms in this study hint at the possibility of multiple strike slip faults in the Hueco Bolson, which can explain the discrepancy observed in the interpreted cross-sections (Figure 5).

Section 2 - Suggestions for Future Work

The SVM algorithm from Scikit-learn allows the user to add weighted values to the input dataset. These weights can be assigned to enforce a higher emphasis on the inputs a geoscientist has a higher confidence in; for example, inputs from well logs, well cores, seismic surveys, or a combination of observations at particular locations. The SVM algorithm can also reproduce the

same models with only the support vectors, reducing the amount of storage space required for these volumetric models. This method can be an alternative to using deep artificial neural networks to compress volumetric models as functions.

In this study I demonstrated how ANN can be effective in reducing the storage space required by a volumetric model by compressing it into a function, resulting in the ANN being the model itself. Further studies need to be conducted on how to modify or extend an ANN to solve geophysical inverse problems. Another possible avenue lies in the use of an ANN as an input for classical methods of forward modeling such as finite differences, finite elements, or finite volumes.

References

Abbate, E., P. Passerini, and L. Zan, 1995, Strike-slip faults in a rift area: A Transect in the Afar Triangle, East Africa. Tectonophysics, 241, no. 1-2, 67-97.

Avila, V.M., D. Doser, O.S. Dena-Ornelas, M.M. Moncada, and S. S. Marrufo-Cannon, 2016, Using geophysical techniques to trace active faults in the urbanized northern Hueco Bolson, west Texas, USA, and northern Chihuahua, Mexico, Geosphere, 12, 264-280.

Baldwin, B., 1956, The Santa Fe Group of northern-central New Mexico. New Mexico Geological Society Guidebook 7, 115-121.

Chiaramonte, M.M, and M. Kiener, 2013, Solving differential equations using neural networks http://cs229.stanford.edu/proj2013/ChiaramonteKiener-SolvingDifferentialEquationsUsingNeuralNetworks.pdf.

Collins, E. W., and J. A. Raney, 1991, Tertiary and Quaternary structure and paleotectonics of the Hueco Basin,Trans-Pecos Texas and Chihuahua, Mexico: The University of Texas at Austin, Bureau of Economic Geology, Geological Circular 91-2, 44 pp. doi.org/10.23867/gc9102D.

Gates, J. S., D. E. White, W.D Stanley, and H.D. Ackermann, 1980, Availability of fresh and slightly saline groundwater in the basins of westernmost Texas: Texas Department of Water Resources Report 256, 108 pp.

George, P.G., R.E. Mace, and R. Petrossian, 2011, Aquifers of Texas: Texas Water Development Board, Report 380.

Guyon I. B., B. Boser, and V. Vapnik, 1993, Automatic Capacity Tuning of Very Large VC-dimension Classifiers. Adv. Neural Inf. Process Sys., 147-155.

Hampson, G.J, M.D Jackson, P.J.R Flitch, and C.M. John, 2012, Surface Based Reservoir Modelling: Concepts and Application to Carbonate Reservoirs. Search and Discovery Article 120060

Hawley, J.W., J.R. Kennedy, A. Granados-Olivas, and M.A. Ortiz, 2009, Hydrologic Framework of the Binational Western Hueco Bolson-Paso del Norte Area, Texas, New Mexico, and Chihuahua: Overview and Progress Report on Digital-Model Development: New Mexico Water Resources Research Institute Technical Completion Report 349, 45 pp.

Hibbs, B. J., R. N. Boghici, M. E. Hayes, J. B. Ashworth, A. T. Hanson, Z. A. Samani, J. F. Kennedy, and B. J. Creel, 1997, Transboundary aquifers of the El Paso/Ciudad Juarez/Las Cruces region. Texas Water Development Board and New Mexico Water Resources Research Institute Report. U.S. Environmental Protection Agency, Dallas, Texas, USA.

Kingma, D. P., and J. L. Ba, 2015, Adam: a Method for Stochastic Optimization. International Conference on Learning Representations, 1–13.

Mace, R.E, W.F. Mullician III, and E.S Angle, 2001, Aquifers of West Texas, Texas Water Development Board, Report 356.

Morgan, P., W.R. Seager, and M.P. Golombek, 1986, Cenozoic Thermal, Mechanical and Tectonic Evolution of the Rio Grande Rift, JGR: Solid Earth, Volume 91, Issue B6, Pages 6263-6276.

Olsen, K.H, W. S. Baldridge, and J. F. Callender, 1987, Rio Grande rift: An Overview: Tectonophysics, 143, no. 1-3, 119-139.

Pedregosa et al., 2011, Scikit-learn: Machine Learning in Python, JMLR 12, 2825-2830.

Smirnoff, A., E. Boisvert, and S.J. Paradis, 2008, Support Vector Machine for 3D modelling from sparse geological information of various origins: Computers and Geosciences 34, 127-143.

Sheng, Z., R. E. Mace, and M.l P. Fahy, 2001, Texas Water Development Board. The Hueco Bolson - An Aquifer at the Cross Roads. Aquifers of West Texas, Edition: Report 356, Chapter 6.

Sheng, Z., M. Darr, J.P. King, J. Bumgarner, A. Michelsen, 2013, Mesilla Basin/Conejos-Médanos Section of the Transboundary Aquifer Assessment Program, in Five-Year Interim Report of the United States – Mexico Transboundary Aquifer Assessment Program: 2007 – 2012, edited by William M. Alley.

Specht, D.F, 1990, Probabilistic Neural Networks, 3, 109-118,

Urbanczyk, K., J. C. White , and D. R. Rohr, 2001, Texas Water Development Board, Report 356, Chapter 2. Geologic History of West Texas, 17 – 25.

U.S. Census Bureau, 2010, Estimates of Total Housing Units by State and County, census.gov.

Vapnik, V., 1995, The Nature of Statistical Learning Theory: Springer-Verlag, New York, 311 pp.

Xu, J., W, Zhongbin, T. Chao, L. Xinhua, 2016, A State Recognition Approach for Complex Equipment Based on a Fuzzy Probabilistic Neural Network, Algorithms, 9, 34, 10.3390/a9020034.

Vita

Joel Castro is a computational geoscientist with an interest in applying numerical modeling, optimization, machine learning, and deep learning to geology and geophysics problems. He earned a bachelors of science in geophysics and a masters of science in computational science from The University of Texas at El Paso. He has worked in the environmental and mineral exploration industry as a geophysicist; conducting various geophysical surveys and interpreting the results. He has also worked in the oil and gas industry as a software engineer; developing a desktop application to diagnose ExxonMobil's next generation reservoir simulator, and developing a RESTful web API for ExxonMobil's subsurface database.

Contact Information: joey-92@live.com