Application of Machine Learning Algorithms to Mineral Prospectivity Mapping

by

Justin Granek

B.Sc., Acadia University, 2009
M.Sc., The University of British Columbia, 2012

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies
(Geophysics)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)
December 2016
© Justin Granek 2016
Abstract

In the modern era of diminishing returns on fixed exploration budgets, challenging targets, and ever-increasing numbers of multi-parameter datasets, proper management and integration of available data is a crucial component of any mineral exploration program. Machine learning algorithms have successfully been used for years by the technology sector to accomplish just this task on their databases, and recent developments aim at appropriating these successes to the field of mineral exploration. Framing the exploration task as a supervised learning problem, the geological, geochemical and geophysical information can be used as training data, and known mineral occurrences can be used as training labels. The goal is to parameterize the complex relationships between the data and the labels such that mineral potential can be estimated in under-explored regions using available geoscience data.

Numerous models and algorithms have been attempted for mineral prospectivity mapping in the past, and in this thesis we propose two new approaches. The first is a modified support vector machine algorithm which incorporates uncertainties on both the data and the labels. Due to the nature of geoscience data and the characteristics of the mineral prospectivity mapping problem, uncertainties are known to be very important. The algorithm is demonstrated on a synthetic dataset to highlight this importance, and then used to generate a prospectivity map for copper-gold porphyry targets in central British Columbia using the QUEST dataset as a case study.

The second approach, convolutional neural networks, was selected due to its inherent sensitivity to spatial patterns. Though neural networks have been used for
Abstract

mineral prospectivity mapping, convolutional neural nets have yet to be applied to the problem. Having gained extreme popularity in the computer vision field for tasks involving image segmentation, identification and anomaly detection, the algorithm is ideally suited to handle the mineral prospectivity mapping problem. A CNN code is developed in Julia, then tested on a synthetic example to illustrate its effectiveness at identifying coincident structures in a multi-modal dataset. Finally, a subset of the QUEST dataset is used to generate a prospectivity map using CNNs.
Preface

This thesis contains original research conducted while studying at the University of British Columbia, resulting in three publications and a conference extended abstract.

The idea of using mineral prospectivity mapping for exploration targeting came from experience working with real datasets for mineral exploration at Computational Geosciences and through discussions with Dr. Eldad Haber, Dr. Elliot Holtham, Dr. David Marchant and Livia Mahler. The decision to start with support vector machines was based on investigation of industry best practices in machine learning and experience trying various algorithms on a preliminary dataset.

The decision to write our own SVM codes was motivated by the lack of currently available packages to incorporate uncertainties on the data and labels, which from experience working in geophysical inversion was known to be extremely important when working with geoscience data. The subsequent Matlab codes, including the adapted SVM formulation published in J. Granek and E. Haber, “Advanced Geoscience Targeting via Focused Machine Learning Applied to the QUEST Project Dataset, British Columbia.” in Geoscience BC Summary of Activities 2015, 2015:117-125, as well as the new $L_1$-RSVR formulation published in J. Granek and E. Haber, “Data mining for real mining: A robust algorithm for prospectivity mapping with uncertainties”, in Proceedings of the 2015 SIAM International Conference on Data Mining, 2015:145-153, were developed and written in collaboration with Dr. Eldad Haber.

The idea of applying convolutional neural networks to the mineral prospectivity problem came from discussions with Dr. Eldad Haber and was sparked by their
Preface

current popularity in the machine learning community. The decision to write our own CNN package in Julia came from a frustration in understanding the inner workings of existing packages and a desire to have control over all aspects of the algorithm. The subsequent code was written by myself with heavy input and assistance from Dr. Haber. This work has been submitted for publication as J. Granek and E. Haber, “Deep Learning for Natural Resource Exploration: Convolutional Neural Networks for Prospectivity Mapping”, *IEEE Transactions on Geoscience and Remote Sensing*, and is currently going through the review process.

Acquisition, processing and sampling of the field data for the QUEST region were performed by myself using open source Quantum GIS software, with consultation from staff at NEXT Exploration.

A summary of this research was presented at the ASEG 2016 conference in Adelaide, and published in the extended abstracts as J. Granek, E. Haber and E. Holtham, “Resource Management Through Machine Learning”, *ASEG Extended Abstracts*, 2016:1-5.
Table of Contents

Abstract ................................................................. ii
Preface ................................................................. iv
Table of Contents ................................................... vi
List of Figures ......................................................... ix
Acknowledgements ................................................... xiii
Dedication ........................................................... xiv

1 Introduction .......................................................... 1
  1.1 Motivation ....................................................... 1
  1.2 Research Outline ................................................ 3
  1.3 Research Summary ............................................... 6

2 Mineral Prospectivity Mapping ................................. 8
  2.1 General Formulation ............................................ 8
  2.2 Previous Work .................................................. 9
  2.3 Problem Characteristics ....................................... 10

3 Support Vector Machines ........................................ 15
  3.1 Maximum Margin Classifier .................................... 15
# Table of Contents

3.2 Separable Linear SVM ................................................. 16
3.3 Variants and Modifications .......................................... 19
  3.3.1 Inseparable SVM ............................................... 20
  3.3.2 Non-linear SVM ............................................... 21
  3.3.3 Multiclass SVM ............................................... 23
  3.3.4 Support Vector Regression .................................... 24
3.4 Robust SVR with Uncertainty ...................................... 27
  3.4.1 Total Least Squares .......................................... 28
  3.4.2 Total Least Norm - $L_1$ Sparsity ........................... 29
  3.4.3 Case of Different Variances ................................ 32
3.5 Synthetic Data Example .............................................. 34

4 Convolutional Neural Networks ...................................... 38
  4.1 Neural Networks .................................................. 38
    4.1.1 History .................................................... 38
    4.1.2 General Architecture ...................................... 40
    4.1.3 Optimization ............................................... 44
  4.2 Convolutional Neural Networks .................................. 45
    4.2.1 Convolutional Layer ...................................... 45
    4.2.2 Pooling Layer ............................................. 47
    4.2.3 Classification/Objective Function ........................ 47
    4.2.4 CNN Architecture .......................................... 48
  4.3 Synthetic Data Example .......................................... 49

5 QUEST Project Field Example ....................................... 56
  5.1 QUesnelia Exploration STTrategy Data .......................... 56
  5.2 The Target: Porphyry Copper-Gold Systems ..................... 60
  5.3 Exploration Goals .............................................. 63
# Table of Contents

5.4 Data Preparation ................................................. 63
5.5 SVM Prospectivity Mapping ................................. 67
  5.5.1 Problem Setup ............................................. 67
  5.5.2 Results .................................................... 67
5.6 CNN Prospectivity Mapping ................................. 70
  5.6.1 Problem Setup ............................................. 70
  5.6.2 Results .................................................... 70

6 Conclusions ....................................................... 73
  6.1 Major Contributions .......................................... 73
  6.2 SVM vs CNN Comparison ...................................... 77
  6.3 Remaining Challenges & Future Work ....................... 79

Bibliography .......................................................... 82
List of Figures

2.1 Mineral prospectivity mapping aims to combine numerous geoscience layers to derive a map of mineral potential. ........................ 14

3.1 Optimal margin classification using SVM. The seperating hyperplane (black) is determined by maximizing the margin (yellow) between a few sparse support vectors (outlined in green). ......................... 16

3.2 Calculating the distance between the hyperplane and a nearest point in feature space, X_n. ........................................... 17

3.3 Inseparable SVM: points are allowed to be misclassified, but are penalized using positive slack variables \( \xi \). ............................................ 20

3.4 Non-linear transformation via the kernel trick. The data is linearly separable in the transformed feature space. ......................... 22

3.5 Multiclass SVM problem can be solved as one-vs-rest or as one-vs-one to generate multiple decision boundaries. ...................... 23

3.6 Support vector regression using \( \epsilon \)-insensitive loss function; only values greater than \( \pm \epsilon \) contribute to the objective value (see equation 3.18). 24

3.7 Three of the most common loss functions used in conjunction with SVMs. Note: The hinge loss is using \( \epsilon = 0.5 \). ......................... 26
List of Figures

3.8 Full synthetic dataset. Separable binary dataset is rendered inseparable by two forms of errors: bad data (purple boxes) and bad labels (green boxes). Training data are circles and predicted labels are triangles. Marker size is proportional to label confidence. ........................................ 35

3.9 Prediction results from synthetic example (Left: $L_1$-RSVR and right: SVR without uncertainties). Training data are circles and predicted labels are triangles. Marker size is proportional to label confidence. Separating boundaries shown in green and pink, and black crosses indicate mis-classified points. ......................................................... 36

4.1 Neural networks are modelled after biological neurons in the brain, in which each neuron will only fire (perpetuate a signal) if the total input signal exceeds some threshold (taken from Stanford CS231n course material). ................................................................. 39

4.2 Example diagram of a feed-forward neural network. This network has 3 inputs and 3 layers: two hidden layers with 4 neurons each, and one output layer (taken from Stanford CS231n course material). ........ 41

4.3 Different activation functions commonly used in neural networks. . . . 41

4.4 Supervised vs unsupervised neural networks. Top: Feed-forward neural networks uses training labels to optimize the hidden layer parameters; bottom: Autoencoders minimize the difference between the training data and the output feature vector to find optimal compressed representation in the hidden layer. ......................... 43

4.5 Backpropagation of error through a neural network ....................... 45

4.6 Diagram of a typical convolution layer in a convolutional neural network (taken from DeepLearning.net's tutorial on CNN) .......... 46
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7</td>
<td>Diagram of a typical max-pooling operation in a convolutional neural network (taken from Stanford CS231n course material).</td>
<td>47</td>
</tr>
<tr>
<td>4.8</td>
<td>Example architecture of a convolutional neural network, stacking (two) layers of convolution and pooling with a final fully connected layer (taken from DeepLearning.net’s tutorial on CNN).</td>
<td>49</td>
</tr>
<tr>
<td>4.9</td>
<td>A single example from the synthetic dataset. a) First channel is a smoothly decaying ellipsoid with a randomly chosen origin and radii, b) second channel is a smoothly decaying ring, also with randomly chosen origin and radii, and c) third channel is a discrete block with random origin and dimensions.</td>
<td>50</td>
</tr>
<tr>
<td>4.10</td>
<td>Examples of a) positive and b) negative training data. Notice how all three channels overlap in the positive example, but not in the negative example.</td>
<td>51</td>
</tr>
<tr>
<td>4.11</td>
<td>Convergence plots for the best result running the synthetic dataset in the CNN. Blue curve is training data, red curve is validating.</td>
<td>53</td>
</tr>
<tr>
<td>4.12</td>
<td>False negative classification results from synthetic dataset. Contour plots of channels 1,2,3 are plotted as red, blue and green, respectively.</td>
<td>54</td>
</tr>
<tr>
<td>4.13</td>
<td>False positive classification results from synthetic dataset. Contour plots of channels 1,2,3 are plotted as red, blue and green, respectively.</td>
<td>55</td>
</tr>
<tr>
<td>5.1</td>
<td>QUEST project area in central British Columbia.</td>
<td>56</td>
</tr>
<tr>
<td>5.2</td>
<td>Porphyry belts in central British Columbia.</td>
<td>57</td>
</tr>
<tr>
<td>5.3</td>
<td>Sample of the available geoscience maps in the QUEST region.</td>
<td>59</td>
</tr>
<tr>
<td>5.4</td>
<td>Major porphyry systems around the world, taken from (John et al., 2010).</td>
<td>60</td>
</tr>
<tr>
<td>5.5</td>
<td>Illustrative example of different signatures over a copper-gold porphyry deposit (Alumbrera deposit in Argentina), taken from (Hoschke, 2010).</td>
<td>61</td>
</tr>
</tbody>
</table>
5.6 Diagram of a copper porphyry system, taken from (Sillitoe, 2010).

5.7 **Left:** Known mineralization locations in the validation set of the QUEST project area. **Right:** Uncertainty estimates for the mineralization labels (red is more uncertain) in the validation set.

5.8 Prediction for prospective mineral regions in validation set of the QUEST area (red is more prospective, blue is less) from SVM.

5.9 Prediction for prospective mineral regions in QUEST area (red is more prospective, blue is less) from SVM using only potential field data.

5.10 Convergence plots for the best result running the QUEST dataset in the CNN. Blue curve is training data, orange curve is validating.

5.11 Mineral prospectivity map generated from CNN for QUEST region. Colorscale maps the probability of mineralization from 0-1 (blue-red). Known mineral occurrences are plotted as black dots for reference.
Acknowledgements

Most importantly, I must thank my supervisor Eldad Haber for his mentorship over the last 4 years. This degree, and likely my career path, would not exist if it were not for you. I am grateful for all the knowledge, guidance, kindness and arguments you have shared with me during the course of my program. Your willingness to explore new topics and your dedication to understanding them have kept my research exciting and fresh, and left me with a strong foundation for future endeavours.

A special thanks goes out to NEXT Exploration for sponsoring my research through a Mitacs Accelerate Fellowship. Thanks also to Geoscience BC for the generous funding through their scholarship program, and to UBC for additional support.

Thanks to the team at Computational Geosciences for the experience of working in industry and creating the motivation to complete this degree. In particular thanks to Elliot Holtham, Dave Marchant and Livia Mahler for the support through-out my PhD, both in tackling the field data from the QUEST project and in keeping me driven to finish.

To all my office mates and colleagues at UBC-GIF, thank you for the assistance and fun over the past 7 years! Gudni, Lindsey, Mike and Sarah, it’s been a pleasure bouncing ideas off of you and moving through our degrees together.

To all my friends and family outside the UBC bubble that have been with me on this journey and have kept me sane and focused, your support and friendship are valued beyond measure. In particular thank you to my parents for your love and guidance and for always being there to pick me up and push me onwards.
To my parents.
Chapter 1

Introduction

1.1 Motivation

Over the last 25 years the number of major mineral discoveries has drastically decreased, as have the profit margins in the mining sector (Barnett and Williams, 2006). This finding is counter-intuitive given the rising global population, rapid industrialization of emerging economies and the shrinking globe. As with most fields, one would expect as the steady march of science and progress give rise to advances in technology and methodology efficiency should increase; however in mineral exploration this has not been the case.

This shortage in new mineral resource discoveries is partially due to inertia of the industry in adopting new technologies, and partially a simple phenomenon of limited natural resources. The majority of current active mines were discovered because they either had outcropping mineralization (geology) or were relatively shallow bodies exhibiting fairly strong geophysical or geochemical signatures. These represent the low hanging fruit in the context of exploration targeting; the remaining targets will not be so easily discovered. Future mineral prospects will likely bear at least one of the following challenges:

- Deeper ore bodies
- Complex mineral systems with non-trivial signatures in the data
- Obscured beneath overburden or other complex geology
1.1. Motivation

This is not to suggest that the mining industry is ill-prepared to explore these targets. On the contrary, advances in geoscience have been consistent, producing a wide range of new tools for prospecting. One such advance has been the development of techniques and technologies for the integration of large quantities of geoscientific data via mineral prospectivity mapping.

Traditional exploration teams would operate within the bounds of their respective fields (ie: geology, geochemistry and geophysics), with little or no collaboration between the disciplines. The challenge with this approach is that in many cases the whole is greater than the sum of parts when targeting a mineral deposit, and the exploration program can benefit immensely from a coordinated approach involving a diverse team of experts. Because of this, integrated exploration targeting teams have become much more the norm over the last decade.

Despite this, one of the remaining challenges resides in objective, efficient handling of the ever increasing volume of available geoscientific data. In many developed countries exploration data sets are made public, either by local government agencies or as a mandatory clause for all data collection (Bastrakova and Fyfe, 2012). This array of knowledge and data can be staggering, spanning various aspects of geology, geochemistry and geophysics. While indisputably valuable, expert analysis of geoscientific data is always subject to various degrees of personal bias and interpretation, and does not always offer a consistent understanding of an exploration target. This can be particularly pronounced when attempting to integrate interpretations from different disciplines, such as geophysical inversion models and geological cross sections. Given that all the data is being generated by sampling some function of the same rocks - whether via airborne magnetometer or analysis of outcrop in hand samples - it stands to reason there should be some correlation between the various data types that ties them all together.
1.2 Research Outline

Mineral prospectivity mapping (see Chapter 2) aims to tackle this challenge by leveraging the intuitive integration possible using geographical information systems (GIS) in conjunction with the computational power of optimization algorithms borrowed from the machine learning community. These algorithms have enjoyed much success over the last 25 years in fields such as pattern recognition, bioinformatics, fraud detection, and many others. As advanced data analysis tools, the power in machine learning algorithms is in their ability to highlight hidden insights in datasets without the need to explicitly search for them. Starting from some assumed flexible model $f(x; \theta)$ of the data $x$ with tuneable parameters $\theta$, machine learning algorithms will learn the correct representation of the data by iteratively updating the parameters to minimize some objective function, such as the residual sum of squares (RSS)

$$\text{RSS} = \sum_{i=1}^{n} (y_i - f(x_i; \theta))^2$$

which quantifies the goodness of fit for the selected model and parameters with respect to some desired output, $y$. Beginning in the 1980s, various researchers have applied many different machine learning algorithms to mineral prospectivity mapping, ranging from regression to fuzzy logic to neural networks.

As the dataset gets larger and more complicated, more sophisticated algorithms are required. One of the more popular machine learning algorithms from the early 2000s is support vector machines (SVM), in which one tunes a set of weights to obtain a hyperplane in some feature space which optimally separates the data classes (see Chapter 3). SVMs were and are widely used because they have strong theoretical guarantees and they provide an elegant and efficient algorithm to solve for complex non-linear relationships using a sparse classifier. Because of this, the classifier can
1.2. Research Outline

be constructed using a small subset of the training points, meaning the size of the training set has little impact on the complexity of the model.

For these reasons, support vector machines were selected as the starting point for this research. Chapter 3 derives the algorithm from basic principles and outlines some of the common variants, as well as the practical considerations for use. Initial attempts were completed using existing open source SVM software libSVM (Chang and Lin, 2011), however it was quickly apparent that certain characteristics of the mineral prospectivity mapping data (see Section 2.3) required the use of uncertainties in training the algorithm, which was not available in this, or any, SVM package. As a result, SVM algorithms were coded up in MATLAB to incorporate this functionality using both a modified classical SVM formulation (equation 3.21), as well as a new implementation, termed $L_1$-RSVR (robust support vector regression), which begins from a total least squares approach (equation 3.37). To illustrate the effectiveness of the $L_1$-RSVR algorithm at handling uncertainties on both the data and the labels a small toy problem was used as a synthetic example (see Section 3.5).

Satisfied with this success, support vector machines were used to train a classifier on the QUEST dataset (see Chapter 5). This dataset from central British Columbia spans roughly 150,000 km$^2$ and consists of a suite of geoscience data ranging from airborne geophysics, to reintrepreted multi-element geochemistry to geological mapping. The region is known to host copper-gold porphyry systems, a style of economic mineral deposit which occurs in clusters or trends up and down the Pacific coast of the Americas, and accounts for more than 60% of the global copper reserves. Using a database of known mineralization within the QUEST study area, a prospectivity map for copper-gold porphyry systems was generated using the $L_1$-RSVR algorithm (see Section 5.5).

Though validation error was low and results were encouraging, one aspect of the support vector machine algorithm was unsatisfying for this application: the lack
of spatial information. While SVMs are powerful learning machines able to model extremely complex data, they inherently operate point-wise, with no sensitivity to regional or global structure. For the mineral prospectivity mapping problem this can be crucial, since many of the most important indicators of mineralization are spatial patterns rather than anomalous values. Porphyries, for example, are known to exhibit a halo-like structure in both the magnetics as well as certain geochemical signatures, and are known to occur near cross-cutting fault structures. While solutions exist to incorporate spatial information into the point-sample framework of SVMs, this realization presented an opportunity to explore one of the most widely used machine learning algorithms of the last decade: convolutional neural networks (CNNs).

Convolutional neural networks (see Chapter 4) are a subset of neural networks, a popular learning algorithm originally motivated by the hierarchical and distributed manner in which neurons in the brain are able to master complex abstract ideas. First developed in the 1990s, CNNs have gained widespread popularity over the last decade for applications in image, text and speech recognition. Their power comes from the realization that natural images (and sounds) have recurring local structure which can be exploited. Given the recent success of CNNs in image identification, segmentation, and parsing, the algorithm is an ideal candidate for mineral prospectivity mapping in which the task is to identify subtle patterns in a multi-parameter dataset of thematic geoscience maps.

While many open source packages exist for the implementation of convolutional neural networks (ie: Torch, TensorFlow, Caffe, Theano) it was decided that we would code our own so as to have a better understanding of the inner workings of the algorithm. A preliminary CNN code was written in MATLAB, and then a working copy was written in Julia (Bezanson et al., 2014). As the focus of this part of the research was the incorporation of spatial information, uncertainties were left out of the code at this stage. As future work, uncertainties on the training labels can easily
1.3 Research Summary

be added, however uncertainties on the data will require further consideration.

Though many benchmark datasets exist in the machine learning community for vision tasks (ie: MNIST, CIFAR10, ImageNet etc.) none of these highlighted the importance of treating multi-modal data (data which comes from different sensors with different distributions). To address this and test our CNN code we developed a synthetic dataset (see Section 4.3) with multiple input channels as an analog for the mineral prospectivity mapping problem. Content with the results on this synthetic example, a convolutional neural network was trained on a subset of the QUEST dataset to generate a prospectivity map for copper-gold porphyries in central British Columbia (see Section 5.6).

1.3 Research Summary

The main objective of the research presented in this thesis is to explore, develop and apply advanced machine learning algorithms for mineral prospectivity mapping. Accomplishing this requires the synthesis of three existing broad disciplines, each with a large body of work behind them: geoscience, geographical information systems (GIS), and machine learning. This work resulted in the following key contributions:

- A modified support vector machine code was written in Matlab which incorporates uncertainties using extra terms on the traditional formulation

- A support vector regression code ($L_1$-RSVR) was written in Matlab using a new formulation beginning from a total least squares approach to incorporate uncertainties on both data and labels

- A convolutional neural network package was written in Julia using a nonlinear conjugate gradient solver
1.3. Research Summary

- Convolutional neural networks were applied to the mineral prospectivity mapping problem for the first time

- Mineral prospectivity maps were generated for the QUEST dataset in central British Columbia

The rest of this thesis document is organized as follows. Chapter 2 introduces the mineral prospectivity mapping problem, complete with a summary of previous work and the relevant problem characteristics. Chapter 3 derives the robust support vector regression algorithm, $L_1$-RSVR, beginning from a basic explanation of support vector machines and extending them to handle various complexities common with real data, including uncertainties. To highlight the importance of properly incorporating uncertainties, a simple synthetic example is presented.

Chapter 4 builds up the necessary background information for proper understanding and implementation of a convolutional neural network, including a history of their development, as well as a break down of their common components. To demonstrate the utility of CNNs for target identification on multi-modal data, a simple synthetic dataset is presented (different than the example for SVM).

To illustrate the effectiveness of these algorithms on real data, Chapter 5 introduces the QUEST dataset from central British Columbia, including a brief description of the exploration target (copper-gold porphyry systems) and a short discussion on data pre-processing and preparation. Both algorithms, SVM and CNN, are used independently to generate prospectivity maps for the QUEST region.

Finally, in Chapter 6 the work is summarized and the differences in the two approaches are discussed. Remaining challenges and opportunities for future work are highlighted, and the major contributions of this research are stated.
Chapter 2

Mineral Prospectivity Mapping

2.1 General Formulation

Mineral prospectivity mapping takes a wholistic approach to exploration targeting; the basic goal being to recognize and parametrize the subtle patterns and features in the multi-disciplinary dataset which are indicative of a desired mineralization style (ie: copper-gold porphyry). Once this is accomplished, the assumption is that one can then apply the learned model (parameterization) to predict the likelihood of mineralization in other similar exploration environments.

Mathematically, this can be represented as follows. Given a suite $N$ of geoscientific data maps $X_i^{Train}$ for $i = 1..N$ (ie: $X_1$ is geological age, $X_2$ is total magnetic field, etc.) and a database of known mineral occurence locations in the region $y^{Train}$, parameterize a model $f(\theta; X^{Train})$ by tuning $\theta$ to fit the data $X$ to the labels $y^{Train}$ according to some objective function $L$ such that

$$\theta^* = \text{argmin } L(y^{Train} - f(\theta; X^{Train}))$$  

(2.1)

Assuming this parametrization can approximate the relationship between the geoscience data and the known mineral occurences, mineral potential can then be estimated in new areas using available exploration data $X^{Predict}$ as

$$y^{Predict} = f(\theta^*; X^{Predict})$$
2.2. Previous Work

so long as the training and predicting data have the same distributions and are sampled in a similar fashion. In other words, so long as the exploration target and environment are similar for the training and predicting sets, and the data is sampled and processed in the same way, the classifier can be used to predict mineralization in the new area. In general $f(\cdot)$ can be any function, ranging from simple linear classifiers to complex nonlinear functions, and $\theta$ can be any kind of parameterization, ranging from indicator variables to regression weights and more complex representations.

2.2 Previous Work

Mineral prospectivity mapping was first proposed in the late 80s by geoscientists as a statistical method for the integration and interpretation of spatial patterns in geoscience data (Bonham-Carter et al., 1989). The concept was to determine the link between various geoscience datasets (ie: geology/geophysics/geochemistry) and the existence or absence of economic mineralization in an objective way.

As computers became more powerful, accessible, and user-friendly, software such as geographical information systems (GIS) revolutionized the way in which geoscientists could analyze and interpret exploration data. The ability to simultaneously view multiple layers of maps on a common reference system facilitated the development and adoption of sophisticated targeting algorithms for the integration of multiple datasets, resulting in the rise of mineral prospectivity mapping.

One of the original formulations, termed Weights of Evidence (WofE) (Agterberg et al., 1990), used posterior probability as the mapping function, which is calculated by counting the relative number of occurrences within and without a series of binary thematic layers (ie: Granite Contact). Other authors have explored using logistic regression (Harris and Pan, 1999), wherein weights are calculated for a series of geological variables using least-squares regression on the probability of mineralization.
2.3. Problem Characteristics

and the binary presence or absence of mineralization. To address the inherent uncertainty in geoscience data, handle categorical variables, and incorporate some expert knowledge, some authors have used fuzzy logic (Porwal et al., 2003b), in which a membership function acts in the place of uncertainty to quantify the degree to which statements are true.

In the last 15 years more sophisticated algorithms have been borrowed from the machine learning field. Feed forward neural networks (see Section 4.1), now ubiquitous in all forms of artificial intelligence, had humble beginnings in the early 1990s and by the 2000s were being applied in simple architectures for mineral prospectivity mapping (Barnett and Williams, 2009; Brown et al., 2000; Harris and Pan, 1999; Porwal et al., 2003a; Rodriguez-Galiano et al., 2015; Singer and Kouda, 1997). Support vector machines (see Section 3.2), a maximum margin classifier which rose to prominence in the 1990s as a favoured algorithm with strong theoretical background, has been applied by a number of authors (Abedi et al., 2012; Porwal et al., 2010; Rodriguez-Galiano et al., 2015; Zuo and Carranza, 2011).

Due to the difficulty in field testing algorithms for this application and the relatively slow adoption of these methods by industry, most of the work on mineral prospectivity mapping has been academic. That being said, some of the more commonly adopted methods include weights of evidence, fuzzy logic and neural networks (Partington and Sale, 2004; Raines et al., 2010). The popularity of these methods can be attributed to ease of use, flexibility, and successful application in other fields.

2.3 Problem Characteristics

Despite widespread use of statistical and machine learning methods in numerous fields, mineral prospectivity mapping presents a number of challenges due to the following problem characteristics.
2.3. Problem Characteristics

**Few training data**  Although mineral occurrences appear in large well documented trends, the number of occurrences and the size of their footprint relative to the region of exploration is quite small. In British Columbia\(^1\) for example, there are only \(\sim 420\) documented alkalic porphyry occurrences, which when associated with a typical footprint of approximately \(10\text{km}^2\) accounts for less than 1\% of the surface area. In the context of supervised machine learning this is a very small training set, and when one considers that each occurrence will be slightly different the variance within positive training data begins to become important.

**Imbalanced learning problem**  In most supervised machine learning environments, an unbiased training is achieved by approximately sampling uniformly from each class. When this is not true, the problem is termed *imbalanced* (Chawla et al., 2004), and can lead to poor generalization of the resulting predictor. A number of methods exist to handle imbalanced data, including boosting (Chawla et al., 2011; Guo and Viktor, 2004) and re-balancing (Kubat and Matwin, 1997; Raskutti and Kowalczyk, 2004; Tang et al., 2009). As might be expected, mineral occurrences are relatively rare, resulting in an extremely imbalanced set of training labels. This problem is further exacerbated when one restricts the problem to a specific type of mineralization (ie: VMS deposits), as is often the case for prospectivity studies.

**Uncertain training labels**  On top of the imbalanced nature of the mineral prospectivity mapping problem is the large degree of uncertainty associated with the training labels. In this regard, there are two fundamental problems: 1) The crucial distinction that in most cases a label of “no mineralization” simply means that mineralization has not been discovered, and not necessarily that there is none, and 2) within each class (mineralized and not mineralized) exists a large range in certainty. For exam-

---

\(^1\)British Columbia is part of a large well-known porphyry belt which extends from Alaska down to Chile (Sinclair, 2007).
2.3. Problem Characteristics

For example, in many mineral occurrence databases “mineralization” encompasses occurrences ranging from producing mines all the way down to anomalies and prospects. One can also understand how a classification of “no mineralization” bears very different implications in the middle of a highly explored mining district than it does in a remote location miles from the nearest field sample site.

**Uncertain training data** As with any observed data, the training data in the mineral prospectivity mapping problem has associated with it uncertainty from various sources. Some data, such as a magnetic total field measurement, will have numerical uncertainties associated with detection limits and processing procedures. Others, such as geological mapping of bedrock units, will have qualitative uncertainties associated with expert interpretation and sampling bias. Additionally, some data can have a spatially correlated uncertainty introduced by different exploration environments in the field (ie: under regions of thick overburden it becomes prohibitively difficult to map bedrock). Unlike many machine learning problems where both the data and the labels can be trusted, for mineral prospectivity mapping it is known that both have associated errors. Furthermore, the errors on the different data types can have very large statistical differences.

**Non-trivial multi-disciplinary data** The data typically employed for mineral exploration encompasses a wide variety of sensors and interpretations. Geological mapping of the bedrock is conducted in situ wherever outcrop is available, and interpreted to produce multiple lithological maps of the regional and local geology, including parameters such as age, rock type, primary minerals and fault structure. Geochemical sampling of soils and sediments can produce maps of elemental content in ppm or % which can be instrumental in identifying and interpreting alteration zones and other processes indicative of mineralization. Geophysical surveys provide
2.3. Problem Characteristics

means to peek beneath the surface in a non-invasive way. Various surveys are able to sample the gravitational, magnetic, or electromagnetic fields in the ground to produce maps or volumetric models which can be correlated to the physical properties of the rocks. Since mineral occurrences are anomalous bodies of ore, they often bear geophysical signatures which are identifiable (Ford et al., 2004).

Mineral systems are complex Given the variety in the multi-sensor data, the expected structure in each input differs greatly, however it is the co-occurrence of a number of these structures which is indicative of prospective mineralization (i.e.: a magnetic high on its own can be meaningless, however when associated with the correct geology, conductivity and geochemical signature, can be very compelling as a potential target).
2.3. Problem Characteristics

Figure 2.1: Mineral prospectivity mapping aims to combine numerous geoscience layers to derive a map of mineral potential.
Chapter 3

Support Vector Machines

3.1 Maximum Margin Classifier

Support Vector Machines (SVMs) were first formally introduced in the 90s by Vapnik, Boser and Guyon (Boser et al., 1992; Cortes and Vapnik, 1995; Vapnik, 1995, 1999) as a machine learning algorithm structured on the statistical learning theory (VC theory) developed by Vapnik and Chervonenkis during the 60s and 70s (Vapnik and Chervonenkis, 1971; Vapnik and Lerner, 1963). The basic principle of SVMs is to construct an optimal margin classifier which has complexity based not on the dimensionality of the feature space, but rather on the number of support vectors, thus allowing for sparse solutions (see Figure 3.1). This can be a powerful advantage when dealing with large highly dimensional datasets since only a small subset of data are necessary in constructing the SVM classifier. SVM falls under the branch of machine learning known as supervised learning, in which a predictor is learned using training data and training labels (Friedman et al., 2001). In the simplest case, one can consider training data with binary labels

\[(X_1, y_1), (X_2, y_2), ..., (X_n, y_n)\] with \(X_i \in \mathbb{R}^m, y_i \in \{-1, 1\}\)

For mineral prospectivity mapping, \(X_i\) would be a row vector of different field measurements (ie: [magnetic total field, bedrock age, distance to fault...etc]) for a given sample location, and \(y_i\) would signify "mineralization" or "no mineralization" for that
3.2 Separable Linear SVM

If the data is linearly separable then one can define a separating hyperplane as

\[ Xw + b = 0 \]  

(3.1)
3.2. *Separable Linear SVM*

where \( \mathbf{w} \) is a column vector of weights and \( b \) is a bias constant. Since the equation of the plane is scale invariant, we can define normalization

\[
|X_n \mathbf{w} + b| = 1
\]  

(3.2)

where \( X_n \) is the nearest point to the hyperplane. To maximize the margin we need to know the distance between the hyperplane and the nearest point, \( X_n \).

This can be calculated by projecting the distance between any point on the hyperplane, \( \mathbf{X} \), and the nearest point \( X_n \) onto the unit vector perpendicular to the plane, \( \hat{\mathbf{w}} \) (see Figure 3.2). Therefore the distance is

\[
\text{Distance} = |(X_n - \mathbf{X})\hat{\mathbf{w}}| = \frac{1}{\|\mathbf{w}\|} |X_n \mathbf{w} - \mathbf{X} \mathbf{w}|
\]  

(3.3)

If we simply add and subtract the bias term \( b \) we get a familiar expression

\[
\text{Distance} = \frac{1}{\|\mathbf{w}\|} |X_n \mathbf{w} + b - \mathbf{X} \mathbf{w} - b| = \frac{1}{\|\mathbf{w}\|} |1 - 0| = \frac{1}{\|\mathbf{w}\|}
\]  

(3.4)
3.2. Separable Linear SVM

Therefore the optimization problem is the following

\[
\text{Maximize } \frac{1}{\|w\|} \quad \text{Subject to } |X_nw + b| = 1 \tag{3.5}
\]

which can be simplified due to our binary label assumption, since for all correctly classified points $|X_iw + b| = y_i(X_iw + b)$. By replacing the maximization with a minimization we arrive at the following constrained optimization problem for separable binary support vector machines

\[
\min_w \frac{1}{2} w^\top w \\
\text{subject to } y_i(X_iw + b) \geq 1 \text{ for } i = 1, \ldots, n. \tag{3.6}
\]

This forms a KKT system which can be solved either iteratively in the primal form (as above) or via quadratic programming in the dual form using Lagrange multipliers $\alpha$

\[
\mathcal{L}(w, b, \alpha) = \frac{1}{2} w^\top w + \sum_{i=1}^{m} \alpha_i \left[1 - y_i (X_iw + b)\right] \nonumber
\]

\[
\mathcal{L}(w, b, \alpha) = \frac{1}{2} w^\top w + \alpha^\top \left[1 - \text{diag}(y) (Xw + b)\right] \tag{3.7}
\]

This Lagrangian can now be minimized with respect to $w$ and $b$, and maximized with respect to $\alpha_i \geq 0$. To do so we can take the derivatives with respect to the primal variables ($w$ and $b$) and set them equal to 0,

\[
\nabla_w \mathcal{L} = w - X^\top \text{diag}(y)\alpha = 0 \tag{3.8}
\]

\[
\nabla_b \mathcal{L} = y^\top \alpha = 0 \tag{3.9}
\]
If we solve equation 3.8 for $w$ and substitute this back into the Lagrangian we get

$$\mathcal{L}(\alpha) = \frac{1}{2} \alpha^\top \text{diag}(y) XX^\top \text{diag}(y) \alpha + \alpha^\top \left[ 1 - \text{diag}(y) \left( XX^\top \text{diag}(y) \alpha + b \right) \right]$$ \hspace{1cm} (3.10)

Expanding and rearranging the terms gives

$$\mathcal{L}(\alpha) =$$

$$\frac{1}{2} \alpha^\top \text{diag}(y) XX^\top \text{diag}(y) \alpha + \alpha^\top - \alpha^\top \text{diag}(y) XX^\top \text{diag}(y) \alpha + (y^\top \alpha) b$$ \hspace{1cm} (3.11)

and since $y^\top \alpha = 0$ the last term drops out. Combining the similar terms, we arrive at the dual form of the linear SVM problem:

$$\text{maximize} \quad -\frac{1}{2} \alpha^\top \text{diag}(y) XX^\top \text{diag}(y) \alpha + 1^\top \alpha$$

subject to $\quad y^\top \alpha = 0 \quad \alpha_i > 0 \text{ for } i = 1..n$ \hspace{1cm} (3.12)

which can finally be rewritten in the common dual form as an unconstrained quadratic optimization problem by noting that maximizing $-f(x)$ is equivalent to minimizing $f(x)$ and adding in the constraint that $y^\top \alpha = 0$ using a regularization parameter $\beta$

$$\text{minimize} \quad \frac{1}{2} \alpha^\top Q \alpha + (-1^\top - \beta y^\top) \alpha$$ \hspace{1cm} (3.13)

where $Q = \text{diag}(y) XX^\top \text{diag}(y)$. Both forms - the primal and the dual - are equivalent, and the solution technique is typically a matter of personal preference, though it can be impacted by the size of the data matrix $X$, since $Q$ is of size $n \times n$.

### 3.3 Variants and Modifications

The optimization problem outlined in the previous section is the simplest to understand, however it obviously makes a number of assumptions which are often violated,
3.3. Variants and Modifications

namely that there are only two classes which are linearly separable. It is easy to extend support vector machines to handle cases in which this is not true, whether because there exist points which violate the margin, the points are separable in a higher dimensional feature space, or the points belong to more than two classes (whether multi-class or continuous values).

3.3.1 Inseparable SVM

Consider the case in which the data is nearly linearly separable, with the exception of a few points (see Figure 3.3). These points are most likely outliers, and should ideally be allowed to violate the margin; the objective should not be overly penalized for them being misclassified.

Figure 3.3: Inseparable SVM: points are allowed to be misclassified, but are penalized using positive slack variables $\xi$.

One way to handle this is with slack variables, $\xi_i$, which quantify the violation for each point. Using this formulation the total violation for all points is $\sum_i \xi_i$ with a
3.3. Variants and Modifications

regularization parameter $C$ and the original SVM equation from (3.6) is modified to

$$\minimize_w \frac{1}{2} w^\top w + C \sum_{i}^n \xi_i$$

subject to $y_i(X_i w + b) \geq 1 - \xi_i$ with $\xi_i \geq 0$ for $i = 1, \ldots, n$  (3.14)

3.3.2 Non-linear SVM

In a more extreme case it is possible that the data are completely inseparable in the original data space. By performing a nonlinear transformation $\Phi : X \rightarrow Z$ one can almost always find a new feature space in which the data is linearly separable (see Figure 3.4).

The key observation for kernel methods (Boser et al., 1992; Hofmann et al., 2008; Schölkopf et al., 1999) such as support vector machines is that to optimize this problem in the dual (Equation 3.15) only the inner product of features is required, $K(X, X') = ZZ^\top = \Phi(X)\Phi(X)^\top$.

$$\minimize_\alpha \frac{1}{2} \alpha^\top Q \alpha + (-1^\top - y^\top) \alpha$$

with $Q = \text{diag}(y)K(X, X')\text{diag}(y)$  (3.15)

This inner product has dimensions $n \times n$, regardless of the dimensionality of the feature space $Z$. This powerful observation means that the data can be transformed to high (even infinite) dimensional space at no cost, allowing support vector machines to be very flexible in fitting complex nonlinear data.

Due to its natural extension to the kernel trick, nonlinear support vector machines are most often solved in the dual, however they can also be solved in the primal via the representer theorem using iterative gradient based methods (Chapelle, 2007; Shalev-Shwartz and Singer, 2011). This can be particularly helpful when the number of data, $n$, is large, since $K(X, X')$ will be a large dense matrix.
3.3. Variants and Modifications

Figure 3.4: Non-linear transformation via the kernel trick. The data is linearly separable in the transformed feature space.

Some of the more common kernel transformations used for support vector machines include polynomial, sigmoidal and radial basis functions, however in principle one can arbitrarily construct their own kernel so long as it obeys Mercer’s condition.
3.3.3 Multiclass SVM

All previous formulations have assumed a binary classification problem. It is easy to think of a situation in which more than two output values are possible. The simplest extension is to apply support vector machines to a classification problem with \( m \) classes.

Figure 3.5: Multiclass SVM problem can be solved as one-vs-rest or as one-vs-one to generate multiple decision boundaries.

Two main strategies exist for tackling this problem. The first, termed one-vs-rest, basically trains \( m - 1 \) classifiers using “one-hot” labels for \( y \) such that

\[
y = \begin{bmatrix} 1 & 1 & 2 & \cdots & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 \ 0 & 0 \ 0 & 1 \ 0 & 0 \cdots & 1 \end{bmatrix}
\]

so that each class is trained versus all others, and membership is assigned for each point to the class which has the largest objective value.
3.3. Variants and Modifications

\[ g(\mathbf{X}) = \sum_{\alpha_n > 0} \alpha_n y_n \mathbf{K}(\mathbf{X}_n, \mathbf{X}) + b \]

The other common strategy is known as one-vs-one, and constructs \( \frac{m(m-1)}{2} \) classifiers (for each pair of classes). In this method the number of times each point is assigned to each class (out of a possible \( m - 1 \) times for each of the \( m \) classes) is counted, and membership is assigned to the class with the maximum.

### 3.3.4 Support Vector Regression

Finally one can envision many applications where the problem isn’t one of classification at all, but rather a regression problem with continuous output values. The most common formulation for this scenario is known as \( \epsilon \)-SVR (Vapnik, 1995), in which the aim is to find an approximating function for the labels which deviates by at most \( \epsilon \) from the target.

![Figure 3.6: Support vector regression using \( \epsilon \)-insensitive loss function; only values greater than \( \pm \epsilon \) contribute to the objective value (see equation (3.18).](image)

Figure 3.6: Support vector regression using \( \epsilon \)-insensitive loss function; only values greater than \( \pm \epsilon \) contribute to the objective value (see equation 3.18).
Mathematically, the optimization problem becomes

$$\min_{w} \frac{1}{2} w^\top w + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \text{ subject to } \begin{cases} y_i - (X_i w + b) \geq \epsilon + \xi_i \\ (X_i w + b) - y_i \geq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$  (3.16)

where $\epsilon_i$ is a measure of the precision of the approximating function and $\xi_i^{(*)}$ are the slack variables allowing for violations of the separating hyperplane. SVR can also be generalized as the following regularized risk minimization

$$\text{Risk}(w) = \sum_{i=1}^{n} \rho(y_i - X_i w) + \frac{\alpha}{2} \|w\|_2^2$$  (3.17)

Where in the case of $\epsilon$-SVR, the loss function is

$$\rho(\sigma; \epsilon) = \begin{cases} 0 & \text{if } |\sigma| < \epsilon \\ |\sigma| - \epsilon & \text{otherwise} \end{cases}$$  (3.18)

In general, for small $\epsilon$, this bares a resemblance to ridge regression, which uses the 1-norm as the cost function ($\rho_{\sigma} = \|\sigma\|_1$). Both of these problems try to minimize some function $\rho_{\sigma}(\cdot)$ of the residual errors ($\sigma$) between the true labels ($y$) and the predicted labels ($\Phi(X)w$).

$$\min_{\sigma,w} \rho_{\sigma}(\sigma) + \frac{\alpha}{2} \|w\|_2^2 \text{ s.t. } \Phi(X)w - y - \sigma = 0$$  (3.19)

As discussed in section 2.3, it is possible however that the labels are not known exactly, and that some uncertainty exists. Furthermore, the same could be true of the data; measurements in the real world often have some associated uncertainty.
3.3. Variants and Modifications

![Figure 3.7: Three of the most common loss functions used in conjunction with SVMs. Note: The hinge loss is using $\epsilon = 0.5$.](image)

One can modify SVR so as to handle not only errors on the labels, but also errors on the data. This has been explored to some extent by previous authors, though in many cases uncertainties are only treated for the either inputs or the outputs; not both (Carrizosa, 2007; Pant et al., 2011; Zhang, 2004). In some instances both uncertainties were simultaneously considered, however the algorithm was treated in the dual form (Huang et al., 2012). As has been previously discussed, this can quickly become cumbersome when dealing with large datasets. One way to handle this would be to modify the primal objective function from (3.6) to incorporate a weighting term proportional to the uncertainty on the labels

$$\Phi(w) = \frac{C_1}{2} w^\top w + \frac{1}{\epsilon} \max(0, 1 - \text{diag}(y) (Xw + b))$$

(3.20)

and a new term which allows for errors in the observed data, $X_{\text{obs}}$, with a user defined uncertainty matrix $\Sigma$ (can be diagonal, full, or anything in between)
3.4 Robust SVR with Uncertainty

\[
\Phi(w, X) = \frac{C_1}{2} w^\top w + \frac{1}{\epsilon} \max(0, 1 - \text{diag}(y) (Xw + b)) \\
+ \frac{C_2}{2} (X - X_{\text{obs}})^\top \text{diag}\left(\frac{1}{\Sigma}\right) (X - X_{\text{obs}})
\]

where \(C_1\) and \(C_2\) are regularization constants which control the trade-off between changing the model, the data fit, and changing the data. This can be solved iteratively for \(w\) and \(X\) until convergence is reached for both unknowns. In the following section an alternative formulation is done using a Total Least Squares approach as a starting point, which lends to an elegant and efficient iterative algorithm for solving the SVR problem with uncertainties on both inputs and outputs.

3.4 Robust SVR with Uncertainty

Contrary to ridge regression/Least Squares where errors are assumed to exist only in the residual between the true labels and predicted labels, Total Least Squares (TLS) allows for errors to exist in the data as well (Golub and Loan, 1980; Golub et al., 1999; Osborne and Watson, 1985). To our knowledge taking this approach to the support vector machine problem has yet to be presented. This results in the following optimization problem:

\[
\min_{\Sigma, \sigma, w} \rho_\Sigma(\Sigma) + \rho_\sigma(\sigma) + \frac{\alpha}{2} ||w||_2^2 \\
\text{s.t.} \quad (\Phi(X) + \Sigma)w - y - \sigma = 0
\]

This is a regularized regression problem that implies that we aim to solve for the weights, \(w\), assuming that the input data \(\Phi(X)\) and the labels \(y\) are inaccurate, with unknown errors \(\Sigma\) and \(\sigma\). We wish to minimize these errors, measured by the loss.
3.4. Robust SVR with Uncertainty

functions $\rho_\Sigma$ and $\rho_\sigma$. The simplest case is total least squares (TLS) where we choose quadratic loss ($\rho_\Sigma(E) = \frac{1}{2}\|E\|_F^2$ and $\rho_\sigma(\epsilon) = \frac{1}{2}\|\epsilon\|^2$) but other loss functions can be used as well. We now quickly review the TLS and then extend it to the l-1 norm.

3.4.1 Total Least Squares

Consider first the case of quadratic loss functions. The Lagrangian is

$$L = \frac{1}{2}\|\Sigma\|_F^2 + \frac{1}{2}\|\sigma\|^2 + \frac{1}{2}\|w\|^2 + \lambda^\top((\Phi + \Sigma)w - y - \sigma)$$

where $\lambda$ is a lagrangian parameter. To obtain the necessary conditions for a minimum we take the derivatives with respect to $w$, $\Sigma$, $\sigma$ and $\lambda$ and set them all equal to zero.

With a little re-arranging we can get $\Sigma = -\lambda w^\top$ and $\sigma = \lambda$. Substituting this relation back in to the optimality conditions we arrive at the following system of two nonlinear equations with two unknowns, $w$ and $\lambda$:

$$\alpha w + \Phi^\top \lambda - w \lambda^\top \lambda = 0 \quad (3.23a)$$
$$\Phi w - \lambda w^\top w - y - \lambda = 0 \quad (3.23b)$$

Defining a new constant $\beta = (w^\top w + 1)^{-1}$ and rearranging we obtain

$$\left(\Phi^\top \Phi + \frac{\alpha - \lambda^\top \lambda}{\beta} I\right)w = \Phi^\top y$$

If we now define another constant $\gamma = \beta^{-1}(\alpha - \lambda^\top \lambda)$ we obtain that the solution to the problem is nothing but the “familiar” Tikhonov regularization

$$(\Phi^\top \Phi + \gamma I)w = \Phi^\top y \quad (3.24)$$
3.4. Robust SVR with Uncertainy

with an unknown regularization parameter $\gamma$. Regardless of the solution strategy, equation (3.24) implies that the weights $w$ are qualitatively similar to the ones obtained when assuming that the modeling error $\Sigma = 0$ and using a regularization parameter that is different than the one used for the discrepancy principle. Notably, this implies that the error

$$\sigma = (\Phi + \Sigma)w - y$$

is not sparse. In the context of SVR a non-zero residual is a support vector and therefore dense residuals imply many support vectors and that may not yield a desirable result in the context of machine learning.

3.4.2 Total Least Norm - $L_1$ Sparsity

To obtain sparse residuals we propose to solve the following optimization problem

$$\min_{\Sigma, \sigma, w} \|\sigma\|_1 + \frac{\beta}{2}\|\Sigma\|_F^2 + \frac{\alpha}{2}\|w\|_2^2$$

s.t. $(\Phi + \Sigma)w - y - \sigma = 0$ (3.25)

Our functional to be minimized contains one main feature: it demands sparse residuals by using $L_1$ on the residual vector $\sigma$, thus obtaining a small number of support vectors. This time, we eliminate $\sigma$ to obtain the unconstrained optimization problem

$$\min_{\Sigma, w} \|(\Phi + \Sigma)w - y\|_1 + \frac{\beta}{2}\|\Sigma\|_F^2 + \frac{\alpha}{2}\|w\|_2^2$$

The Euler Lagrange equations for this problem are

$$0 = (\Phi + \Sigma)^\top \text{sign}((\Phi + \Sigma)w - y) + \alpha w$$

(3.26a)

$$0 = \text{sign}((\Phi + \Sigma)w - y)w^\top + \beta \Sigma$$

(3.26b)
3.4. Robust SVR with Uncertainty

This is a coupled nonlinear system that cannot be decoupled as in the previous case of the total least squares. However, note that for a given $\Sigma$ the system can be thought of as a mixed $L_1/L_2$ recovery problem for $w$ and, for a given $w$ the problem is a mixed $L_1/L_2$ recovery problem for $\Sigma$. In particular, using equation (3.26b) we obtain that

$$\Sigma = -\beta^{-1}\text{sign}((\Phi + \Sigma)w - y)w^T$$

(3.27)

That is, $\Sigma = vw^T$ is a rank one matrix with

$$v = \beta^{-1}\text{sign}((\Phi + \Sigma)w - y)$$

(3.28)

Each of these problems can be solved effectively by a number of methods. Thus, one solution strategy is to use a block coordinate descent, solving for $w$ and $\Sigma$ at each iteration.

**Solving for $w$**

Consider the system (3.26). In the first step we assume that $\Sigma$ is known and solve equation (3.26a) for $w$. This is equivalent to solving the optimization problem

$$\min_w \|\hat{\Phi}w - y\|_1 + \frac{\alpha}{2}\|w\|^2$$

(3.29)

where $\hat{\Phi} = \Phi + \Sigma$. This can be done using iteratively reweighted least squares (IRLS). Using the identity $|t| = t^2/|t|$, $t \neq 0$ and approximating it by $|t| \approx \frac{t^2}{\rho(t)}$ with $\rho(t) = \sqrt{t^2 + \epsilon}$ we rewrite the problem as

$$\min_w (\hat{\Phi}w - y)^\top \text{diag} \left( \frac{1}{\rho(\hat{\Phi}w - y)} \right) (\hat{\Phi}w - y) + \frac{\alpha}{2}\|w\|^2$$

30
3.4. Robust SVR with Uncertainty

Although that at first glance the choice of $\epsilon$ seems to be somewhat unclear, as we will show, we could actually choose $\epsilon = 0$ without any difficulty. In IRLS we solve a sequence of reweighted quadratic problems of the form

$$w_k = \arg \min_w (\hat{\Phi} w - y)^\top \text{diag}\left(\frac{1}{\rho(\hat{\Phi} w_{k-1} - y)}\right) (\hat{\Phi} w - y) + \frac{\alpha}{2} \|w\|^2$$

with the solution

$$\left(2 \hat{\Phi}^\top W^{-1} \hat{\Phi} + \alpha I\right) w_k = \hat{\Phi}^\top W^{-1} y$$ (3.30)

where $W^{-1} = \text{diag}\left(\frac{1}{\rho(\hat{\Phi} w_{k-1} - y)}\right)$. At this point we make an interesting observation. It is well known that the system (3.30) is equivalent to the following

$$-\frac{1}{\alpha} \left(\hat{\Phi}^\top + \frac{\alpha}{2} W\right) z = y$$ (3.31a)

$$\frac{1}{\alpha} \hat{\Phi}^\top z = w$$ (3.31b)

This is sometimes referred to as the “data space system”. Note that in this formulation only $W = \text{diag}\left(\rho(\hat{\Phi} w_{k-1} - y)\right)$ appears and therefore choosing $\epsilon = 0$ does not generate any difficulty.

Solve for $\Sigma$

In this step we assume that $w$ is known and solve equation (3.26b) for $\Sigma$. The equation reads

$$\text{sign}((\Phi + \Sigma) w - y) w^\top + \beta \Sigma = 0$$ (3.32)
This equation implies that $\Sigma$ is a rank one matrix. Setting $\Sigma = vv^T$ for some unknown vector $v$, and substituting into (3.26b) we obtain

$$\left( \text{sign}(\Phi + vw^T)w - y + \beta v \right) w^T = 0$$ \hspace{1cm} (3.33)

and this implies that $\text{sign}(ww^Tv + \Phi w - y) + \beta v = 0$. Setting $\gamma = w^Tw$ and $\hat{y} = y - \Phi w$, and multiplying the equation by $\gamma$ we rewrite the system as

$$\gamma \text{sign}(\gamma v - \hat{y}) + \beta \gamma v = 0$$ \hspace{1cm} (3.34)

We now note that this equation is nothing but the optimality condition to the problem

$$\min_v \|\gamma v - \hat{y}\|_1 + \frac{\beta \gamma}{2} \|v\|^2$$ \hspace{1cm} (3.35)

Furthermore, (3.34) is essentially a system of decoupled nonlinear equations, and therefore, it can be solved analytically for each component

$$v_i = \begin{cases} \frac{\hat{y}_i}{\gamma} & \text{if } \gamma v_i - \hat{y}_i = 0 & -1 \leq \beta v_i \leq 1 \\ \beta^{-1} & \text{if } \gamma v_i - \hat{y}_i < 0 \\ -\beta^{-1} & \text{if } \gamma v_i - \hat{y}_i > 0 \end{cases}$$ \hspace{1cm} (3.36)

The solution to (3.25) can be now obtained by iteratively solving for $w$ (using equations 3.31a and 3.31b) and $\Sigma$ (using equation 3.36 and $\Sigma = vv^T$).

### 3.4.3 Case of Different Variances

A more interesting case that is relevant to our application is when the errors have different variance. This is one way in which estimates of the uncertainties can be incorporated into the MPM problem. In this case problem (3.26) is modified to the
3.4. Robust SVR with Uncertainty

Following

\[
\min_{\Sigma, w} \| \text{diag}(\Gamma)((\Phi + \Sigma)w - y) \|_1 + \frac{\beta}{2} \| \Omega_{\frac{1}{2}} \odot \Sigma \|_F^2 + \frac{\alpha}{2} \| w \|^2
\]  

(3.37)

where \( \Gamma \) is the inverse variance vector of \( \sigma \) (the label error), \( \Omega_{\frac{1}{2}} \) is the square root of the inverse variance of \( \Sigma \) (the data errors) and \( \odot \) is the Hadamard product. In this case the conditions for a minimum are

\[
(\Phi + \Sigma)^\top \text{diag}(\Gamma) \text{sign}(\text{diag}(\Gamma)((\Phi + \Sigma)w - y)) + \alpha w = 0
\]

\[
\text{diag}(\Gamma) \text{sign}(\text{diag}(\Gamma)((\Phi + \Sigma)w - y))w^\top + \beta \Omega \odot \Sigma = 0
\]  

(3.38a)

where \( \Omega = \Omega_{\frac{1}{2}} \odot \Omega_{\frac{1}{2}} \). The first observation to be made is that \( \Sigma \) is no longer a rank one matrix. In fact, if \( \Omega \) is a full rank matrix then so is \( \Sigma \) and \( \Sigma = \Omega^{\odot 1} \odot \mathbf{v} \mathbf{w}^\top \) with

\[
\mathbf{v} = -\beta^{-1} \text{diag}(\Gamma) \text{sign}(\text{diag}(\Gamma)((\Phi + \Sigma)w - y))
\]  

(3.39)

Here we are defining \( \Omega^{\odot 1} \) as the element-wise inverse of the squared variance, \( \Omega \). Once again this is a coupled nonlinear system that can be solved using a block coordinate descent, iteratively solving for \( w \) and \( \Sigma \). Just as before, setting \( \hat{\Phi} = \Phi + \Sigma \) in (3.38a), this is equivalent to solving the following \( L_1/L_2 \) optimization problem

\[
\min_w \| \text{diag}(\Gamma)(\hat{\Phi}w - y) \|_1 + \frac{\alpha}{2} \| w \|^2_2
\]  

(3.40)

which can again be solved using IRLS, with the added weighting of \( \text{diag}(\Gamma) \) in the \( L_1 \) term. Equation (3.38a) can similarly be shown to be equivalent to a simple \( L_1/L_2 \) optimization problem, however due to the Hadamard product it requires a bit of extra algebra and where \( \gamma \) was a scalar in (3.35) it now becomes a diagonal matrix of the
3.5. Synthetic Data Example

form $\gamma = \text{diag}(\Omega (\mathbf{w} \odot \mathbf{w}))$. The optimization problem becomes

$$
\min_{\mathbf{v}} \|\text{diag}(\Gamma) (\gamma \odot \mathbf{v} - \hat{\mathbf{y}})\|_1 + \frac{\beta}{2} \|\gamma \odot \mathbf{v}\|_F^2
$$

(3.41)

where $\hat{\mathbf{y}} = \mathbf{y} - \Phi \mathbf{w}$ remains the same from (3.35) and $\| \cdot \|_F$ is the Frobenius norm.

Once again (3.41) is a system of 1D linear equations and can be solved analytically as

$$
\mathbf{v}_i = \begin{cases} 
\frac{\hat{y}_i}{\gamma_i} & \text{if } \gamma_i \mathbf{v}_i - \hat{y}_i = 0 \text{ and } -1 \leq \frac{\beta \mathbf{v}_i}{\Gamma_i} \leq 1 \\
\frac{\Gamma_i}{\beta} & \text{if } \gamma_i \mathbf{v}_i - \hat{y}_i < 0 \\
-\frac{\Gamma_i}{\beta} & \text{if } \gamma_i \mathbf{v}_i - \hat{y}_i > 0
\end{cases}
$$

(3.42)

3.5 Synthetic Data Example

To demonstrate the impact of incorporating uncertainties into the machine learning algorithm a toy example has been contrived (see figure 3.8 below). The dataset consists of 70 sample points in two dimensions, each with an associated binary label - either -1 or 1 (shown as red or blue). Also associated with each sample are two uncertainties, one on the data and one on the label. To illustrate the effect of uncertainties on the training, the dataset is composed of two separable Gaussian distributions which are made inseparable by the presence of both samples with data errors (purple boxes) and samples with label errors (green boxes).
3.5. Synthetic Data Example

Figure 3.8: Full synthetic dataset. Separable binary dataset is rendered inseparable by two forms of errors: bad data (purple boxes) and bad labels (green boxes). Training data are circles and predicted labels are triangles. Marker size is proportional to label confidence.

The dataset was then split into training and predicting subsets via random sampling of the original data points. Finally, the $L_1$-RSVR algorithm was run on the training data and compared to prediction results from simple SVR without uncertainties (see figure 3.9 below).
3.5. Synthetic Data Example

Figure 3.9: Prediction results from synthetic example (Left: $L_1$-RSVR and right: SVR without uncertainties). Training data are circles and predicted labels are triangles. Marker size is proportional to label confidence. Separating boundaries shown in green and pink, and black crosses indicate misclassified points.
3.5. Synthetic Data Example

From the results it is evident that the incorporation of uncertainties (when they are available) improves the resulting predictor. The only mis-classified points in the $L_1$-RSVR result are the points with uncertain labels (except for one), and the outliers which are assumed to be incorrect, and thus are not penalized for being on the wrong side of the boundary. Though this is only a small contrived toy example, it illustrates how failure to incorporate uncertainties can result in very poor prediction as the classifier tries to fit data which are not to be trusted.
Chapter 4

Convolutional Neural Networks

4.1 Neural Networks

Neural networks are powerful learning algorithms which are now ubiquitous across most fields of artificial intelligence, ranging from image segmentation (Alvarez et al., 2012; Lai, 2015; Long et al., 2015) to text and image identification (Krizhevsky et al., 2012; LeCun and Bengio, 1995; Russakovsky et al., 2014; Simard et al., 2003) to speech and time series analysis (Bengio et al., 2012; Greff et al., 2015; Hochreiter and Schmidhuber, 1997).

4.1.1 History

The term “neural network”, which was coined in the 1950s, is in reference to the algorithms’ attempt to model neural processes in the brain. In the biological setting, synapses are passed from one neuron to the next, and each neuron will only fire if the total input signal exceeds some firing threshold. Beginning from this biological and psychological model, neural networks were developed through the 1940-1960s with major contributions from McCulloch & Pitts (McCulloch and Pitts, 1943), Rosenblatt (Rosenblatt, 1958) and Minsky & Papert (Minsky and Papert, 1969). The most well known of these early models was Rosenblatt’s perceptron, which was in essence a single layer of neurons from a modern neural network using the heaviside step function.
4.1. Neural Networks

Figure 4.1: Neural networks are modelled after biological neurons in the brain, in which each neuron will only fire (perpetuate a signal) if the total input signal exceeds some threshold (taken from Stanford CS231n course material).

for the activation.

\[
f(X; w, b) = \begin{cases} 
1 & Xw + b > 0 \\
0 & \text{otherwise}
\end{cases}
\] (4.1)

The fundamental shortcomings of these early perceptrons were that they were unable to model the “XOR” function and that they were very slow to run. This delayed further advances in neural networks until the 1980s, when the combination of superior computing power, the development of the backpropagation algorithm (Rumelhart et al., 1986), and the realization that stacking multiple layers of perceptrons solved the “XOR” problem.

This first resurgence of neural networks presented them as powerful tools for AI as both theoretical proofs (Hornik et al., 1989) and industrial applications (LeCun et al., 1989) were developed. In particular, the development of convolution neural networks (Cun et al., 1990; LeCun and Bengio, 1995; LeCun et al., 1998) (CNN - see Section 4.2 for more information) for image, text and speech recognition, as well as other methods such as self-organizing maps (SOM) (Kohonen, 1990) and recurrent neural networks (RNN) (Hochreiter, 1998) drove the development and popularity of neural networks through the 1980s and 1990s.
4.1. Neural Networks

As more researchers began turning to machine learning in the 1990s, alternative algorithms began to arise. In particular, support vector machines (Burges, 1998; Cortes and Vapnik, 1995) presented themselves as simpler, more theoretically grounded learning machines which in many cases were performing at least as well as the most carefully crafted neural networks.

It wasn’t until the late 2000s that under the new branding of “deep learning” neural nets made another comeback (Bengio and LeCun, 2007; Glorot and Bengio, 2010; Hinton et al., 2006), due in large part to the increased power and efficiency of modern computers. Since then neural networks have risen to dominate the machine learning community, with networks getting ever deeper (Krizhevsky et al., 2012; Szegedy et al., 2015) as advances in both hardware (Chellapilla et al., 2006; Krizhevsky, 2014) and software (Hinton, 2014; Nair and Hinton, 2010) continue to push them to greater accuracy and success. For a more exhaustive history of their development, the reader is directed to (Schmidhuber, 2015).

4.1.2 General Architecture

Following the original analogy as a network of neurons used to model an artificial brain, neural networks are typically structured in terms of a number of layers and the number of neurons per layer.

They are commonly drawn in diagrams such as Figure 4.2 in which each column of circles represents a layer and each circle represents a neuron. In a fully connected layer the lines connecting each of the neurons from the previous layer to each of the neurons in the current layer represent the weights. Neural networks achieve the non-linear “neural” response through the use of activation functions, typically in the form of some kind of sigmoid (see Figure 4.3).
4.1. Neural Networks

Figure 4.2: Example diagram of a feed-forward neural network. This network has 3 inputs and 3 layers: two hidden layers with 4 neurons each, and one output layer (taken from Stanford CS231n course material).

Figure 4.3: Different activation functions commonly used in neural networks.
4.1. Neural Networks

Mathematically, a single layer of a neural network computes the following non-linear mapping of input features to output features:

\[ \mathbf{X}^{n+1}_i = \sigma(\sum_j \mathbf{X}^n_j w^n_{ij} + b^n_i) \] (4.2)

where \( n \) denotes the layer, \( i \) denotes which neuron in the current layer, \( j \) denotes which neuron in the previous layer, and \( \sigma \) is one of the activation functions from Figure 4.3. By combining multiple neurons together and stacking layers of this architecture neural networks are able to model complex functions using relatively simple operations.

This can be done in both the supervised and unsupervised environment, depending on the architecture and objective function used. Common feed-forward neural networks will often be supervised, employing a loss function such as least squares or logistic regression to compute some metric of the difference between the predicted labels and the true labels \( y \). Alternatively, unsupervised neural networks such as autoencoders use a symmetric architecture to encode, and then decode the input signal, with the goal of recovering the input signal with minimal reconstruction error. Similarly to principle component analysis (PCA), this is essentially computing a new (low-dimensional) basis with which to represent the data.
4.1. Neural Networks

Figure 4.4: Supervised vs unsupervised neural networks. **Top:** Feed-forward neural networks use training labels to optimize the hidden layer parameters; **bottom:** Autoencoders minimize the difference between the training data and the output feature vector to find optimal compressed representation in the hidden layer.
4.1. Neural Networks

4.1.3 Optimization

Regardless of the architecture, training of neural networks involves optimizing the numerous parameters (ie: \( w, b \)) to minimize some objective function. Due to the large number of parameters often used for neural networks (particularly for deep architectures) and the large amount of data required to train them, this is typically done using a batch stochastic gradient descent (SGD) algorithm wherein the gradient of the objective function \( Q(w) \) for the full training set is approximated by the gradient for a (small) subset of the training data, \( m << n \). The reasons given for this are typically the large size of the training data, \( n \) (resulting in a large computational demands for handling the Jacobian), and the online nature of many applications. Many variants of SGD exist which attempt to optimally select the step size, or learning rate, \( \eta \), using a variety of criteria (Duchi et al., 2011), but the basic equation for the parameter update remains the following.

\[
w_{i+1} \rightarrow w_i + \eta \sum_{j=1}^{m} \nabla Q_j(w_i)
\] (4.3)

Though SGD does not solve the optimization problem as accurately as a second-order method such as Gauss-Newton iterations, this can be somewhat justified by the fact that neural networks are non-convex and are known to have many local minima. By stochastically optimizing the parameters with a weaker step direction, one is in effect regularizing the solution.

\[
\frac{\delta Q}{\delta w_i} = \frac{\delta Q}{\delta X_j} \frac{\delta X_j}{\delta X_{j-1}} \ldots \frac{\delta X_i}{\delta w_i} \quad \text{for} \quad j > i
\] (4.4)

Efficient implementation of SGD is done using the back-propagation algorithm (Rumelhart et al., 1986) (Eq. 4.4 above), wherein the derivatives of the objective function can be computed using chain-rule backwards through each of the layers in the network.
4.2 Convolutional Neural Networks

Though first formally developed in the 1990s (LeCun and Bengio, 1995; LeCun et al., 1998), convolutional neural networks gained a considerable amount of popularity over the last decade, to the point where they are ubiquitous in the fields of image, audio and character recognition. As a subset of feed-forward neural networks, they fall under the banner of “deep learning”, in which a number of layers of nonlinear transformations are stacked to produce a higher order feature representation of the original data.

The distinction which sets convolutional neural networks apart is the use of shared weights over a local receptive field. This has two primary effects: 1) the stationarity of natural images is exploited by the local structure of the filter architecture and 2) the number of parameters in the model is vastly reduced. Though many subtle enhancements and modifications exist, the main components of a convolutional neural network are convolution layers, pooling layers, non-linear transformations, and some classification/objective function to be optimized.

4.2.1 Convolutional Layer

A convolution layer in a CNN computes the inner-product between a patch of the input image $X^0_p$ and a kernel, or filter, $K^i_j$. This filter is slided over the entire image...
4.2. Convolutional Neural Networks

to produce a new image, or feature map. Since the kernel is (typically) much smaller than the patch size, this can be represented by a convolution. If all the inner products are collected into a single operation, this results in a Toeplitz matrix of size \([\text{number of output elements, number of input elements}]\) - depending on how boundary conditions are handled for the filter this will vary. Using this large sparse matrix rather than looping renders the entire convolution layer into a single matrix matrix multiplication of the form: \(K_j \ast X^i\) where \(K_j\) is the large Toeplitz matrix of convolution operations and \(X^i\) is a matrix of vectorized inputs concatenated together. Combined with an activation function typically involving a sigmoid (ie: \(\tanh\)) and a bias, \(b_j^i\), this constitutes a “neuron” of the form

\[
Z_j^i = \tanh(K_j^i \ast X^i + b_j^i)
\]

A single layer of a convolutional neural network will consist of a number of different kernels, \(K_j\) being convolved with the same input data \(X^i\) to give a number of different output feature maps \([Z_1^{i+1}, Z_2^{i+1}, ..., Z_K^{i+1}]\) (see figure 4.6).

Figure 4.6: Diagram of a typical convolution layer in a convolutional neural network (taken from DeepLearning.net’s tutorial on CNN).

As the network is optimized, the convolution kernels are “learned” so as to provide the best set of feature maps from which to classify the images. The intuition on why this might be a good idea for mineral prospectivity mapping is quite simple. When
manually analyzing geoscientific data for prospectivity mapping, it is common to work with derived maps (i.e.: vertical derivative of the residual magnetic field) rather than the raw data, since they facilitate the detection and identification of spatial correlations and patterns. By optimizing the CNN, the user is allowing the data itself to guide the selection of which derived feature maps are most useful for the classification problem rather than having to arbitrarily chose them manually.

### 4.2.2 Pooling Layer

Pooling layers in convolutional neural networks serve two purposes. First, they introduce an additional nonlinear transformation to the learning function, in particular by adding a level of local spatial invariance. This is best exemplified by the common “max pooling”, in which case the maximum value from a small patch is selected to represent the entire patch, regardless of its position in that patch. Second, the pooling layers are inherently a down-sampling, resulting in a more compact representation of the feature maps. This condensation of the features allows for fewer parameters in the network, and therefore less computations.

![Diagram of a typical max-pooling operation in a convolutional neural network](taken from Stanford CS231n course material).

### 4.2.3 Classification/Objective Function

The third component in any CNN is a final layer which takes as input the feature maps derived from all previous layers (both convolution and pooling) and outputs a
4.2. **Convolutional Neural Networks**

predicted label (either a value or a probability distribution for the set of label classes). This can be as simple as a set of weights for each element in the final feature maps (regression), as complex as a fully-connected feed-forward neural network (complete with sigmoidal neurons), or any other classification/regression function such as support vector machines.

In order to solve for the numerous parameters in the network, some objective function must be defined which penalizes poor performance during training. In general this is often something of the form

$$\Phi = \| f(\theta; X) - y \|_p$$

where $y$ are the true labels, $f(\cdot)$ is some function of the input data $X$ (the images), $\theta$ are the parameters of the function (in this case this would include all the convolution kernels and biases of the CNN), and $\|\cdot\|_p$ is some $p$-norm of the misfit. The parameters are then iteratively optimized using the back-propagation algorithm [LeCun et al., 1989], which in most cases is some form of (stochastic) gradient descent.

### 4.2.4 CNN Architecture

Putting this all together in subsequent layers, each of which involves convolution operations and pooling operations, produces architectures of the form shown in figure 4.8, where the $j^{th}$ channel of the $(i + 1)$-layer of convolution and pooling can be mathematically expressed

$$X_{j}^{i+1} = P\left(\tanh(K_{j}^{i+1} * X^{i} + b^{i+1}_{j})\right) \quad j = 1..N^{i+1}$$

with $N^{i+1}$ the number of channels, or feature maps, in the $(i + 1)^{th}$ layer as dictated by the chosen number of convolution kernels $K_{j}^{i+1}$, and $b^{i+1}_{j}$ is the bias on the $j^{th}$
4.3 Synthetic Data Example

channel in the \((i+1)^{th}\) layer. \(P(\cdot)\) represents the pooling, or down-sampling, operation. Though it has been shown that a relatively simple two layer neural network is a universal approximator (Hornik et al., 1989), modern networks often employ numerous layers (the latest winners of the ImageNet Large-Scale Visual Recognition Challenge are using upwards of 20 layers in their networks (Szegedy et al., 2015)).

![Figure 4.8: Example architecture of a convolutional neural network, stacking (two) layers of convolution and pooling with a final fully connected layer (taken from DeepLearning.net’s tutorial on CNN).](image)

Figure 4.8: Example architecture of a convolutional neural network, stacking (two) layers of convolution and pooling with a final fully connected layer (taken from DeepLearning.net’s tutorial on CNN).

4.3 Synthetic Data Example

As an illustrative example, a synthetic dataset has been created consisting of three channels. The first channel is a smoothly decaying ellipsoid with a randomly chosen origin and radii, the second channel is a smoothly decaying ring, also with randomly chosen origin and radii, and the third channel is a discrete block with random origin and dimensions (see Figure 4.9).

This data is meant to represent a simplified porphyry exploration model where an overlapping high (channel 1) and halo-type structure (channel 2) with the correct coincident geological structure (channel 3) would be suggestive of mineralization. 500 positive training examples were defined as samples where the structures in all three channels overlapped, and 500 negative selected where they do not (see Figure 4.10).

Since the Julia CNN package uses a nonlinear-conjugate gradient solver rather than the more common stochastic gradient descent, steps are computed for all of the
4.3. Synthetic Data Example

Figure 4.9: A single example from the synthetic dataset. a) First channel is a smoothly decaying ellipsoid with a randomly chosen origin and radii, b) second channel is a smoothly decaying ring, also with randomly chosen origin and radii, and c) third channel is a discrete block with random origin and dimensions

training data rather than mini-batches. The algorithm has a number of exit criteria corresponding to tolerances on different values. A maximum number of iterations is specified, as well as a maximum number of learn search iterations. If the objective function on the training set is not decreased by a sufficient amount, or a sufficiently small value is obtained, convergence is assumed. Additionally, if the derivatives become too small, the optimization will stop since no more progress can be made. Finally, if the validation error (objective function value) increases early stopping is initiated so as to avoid over-fitting and maintain generalization of the solution. All of these exit criteria are tuneable. For these runs the following values were used:

Max iterations 200

Max line search iterations 9

Minimum objective function decrease 0.1%

Objective function target value $0.01 \times N$

Minimum relative derivative size $1e^{-4}$
Figure 4.10: Examples of a) positive and b) negative training data. Notice how all three channels overlap in the positive example, but not in the negative example.
4.3. Synthetic Data Example

The network used for the synthetic problem consists of four layers; the first layer has eleven $5 \times 5 \times 3$ convolution kernels, each of which using a sigmoid nonlinearity \( \frac{1}{1+e^{-x}} \) followed by a $2 \times 2$ softmax pooling function. The second layer has fifteen $5 \times 5 \times 11$ convolution kernels, each of which again using a sigmoid nonlinearity and followed by a $2 \times 2$ softmax pooling function, and the third layer has twenty-seven $3 \times 3 \times 15$ convolution kernels, each of which again using a sigmoid nonlinearity and followed by a $2 \times 2$ softmax pooling function. The final layer is a softmax classifier of the form

\[
y_{j}^{pred}(w_j; X) = \frac{e^{X^\top w_j}}{\sum_{i=1}^{K} e^{X^\top w_i}}
\]

which gives the probability of each data being either a positive example or a negative example (for our example it is binary; in general $K$ can be larger than 2). In total this results in 8,702 parameters in the network.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Success %</th>
<th># Correct</th>
<th>False Positive</th>
<th>False Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>90.4</td>
<td>452/500</td>
<td>18</td>
<td>30</td>
</tr>
<tr>
<td>Validating</td>
<td>93.6</td>
<td>468/500</td>
<td>19</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 4.1: Results from classification of synthetic data. Numbers reported are the best achieved using the user-selected CNN architecture and 20 random initializations.

From 20 randomly initialized runs, the best results were 90.4% successful prediction on the training set and 93.6% successful prediction on a validation set (see Table 4.1). The algorithm converges to a solution in 40 iterations, exiting because the training objective function was no longer changing (see Figure 4.11).

If we look at a subset of the incorrectly classified negative validation data (Figure 4.12), we can see they are almost positive examples; the centers of the 3 channels are close, though not overlapping. Similar but opposite conclusions can in some cases be made when looking at the false positive examples (Figure 4.13), giving some intu-
4.3. Synthetic Data Example

ition and confidence that the CNN algorithm is effectively recognizing the overlapping patterns of the 3 channels.

Figure 4.11: Convergence plots for the best result running the synthetic dataset in the CNN. Blue curve is training data, red curve is validating.
4.3. Synthetic Data Example

Figure 4.12: False negative classification results from synthetic dataset. Contour plots of channels 1, 2, 3 are plotted as red, blue and green, respectively.
4.3. Synthetic Data Example

Figure 4.13: False positive classification results from synthetic dataset. Contour plots of channels 1,2,3 are plotted as red, blue and green, respectively.
Chapter 5

QUEST Project Field Example

5.1 QUesnelia Exploration STrategy Data

The QUEST (QUesnelia Exploration STrategy) project was a large $5M initiative by GeoscienceBC and the local government to encourage mineral exploration in central British Columbia. The region, spanning approximately 150,000 km$^2$, sits within established porphyry belts which extend the length of the province (see Figure 5.2), and is known to host a number of economic copper/gold porphyry deposits (Devine, 2012), however many of the easier surface targets have already been found.

Figure 5.1: QUEST project area in central British Columbia
5.1. QUesnelia Exploration STrategy Data

Figure 5.2: Porphyry belts in central British Columbia

To stimulate further discoveries, an extensive data acquisition program was completed from 2008-2012, including airborne geophysics (Barnett and Kowalczyk, 2008), geochemical sampling (Jackaman and Balfour, 2008), and extensive geological field mapping. The result was a large, rich database of multi-parameter geoscience data.
5.1. Quesnelia Exploration Strategy Data

in a region of known mineral potential for copper/gold porphyries. In addition to the newly acquired data, historical data was compiled and re-analyzed, including a database of known mineral occurrences.

This data rich environment has been used as a case study by many geoscientists for various projects, including mineral prospectivity mapping (Barnett and Williams, 2009; Granek and Haber, 2015). Amongst others, the publicly available datasets in the QUEST area include airborne gravity, magnetic and electromagnetic data, inductively coupled plasma mass spectrometry (ICP-MS) analysis of stream and sediment samples (providing compositional information for 35 elements), geological era, period, rock class and rock type, and location and classification of known mineral occurrences.
5.1. Quesnelia Exploration Strategy Data

Figure 5.3: Sample of the available geoscience maps in the QUEST region.
5.2 The Target: Porphyry Copper-Gold Systems

Copper-gold porphyry systems are important exploration targets. They account for over 60% of the global copper production, as well as a substantial percentage of global gold, molybdenum and to a lesser extent silver (John et al., 2010). Though they can be lower grade mineralization, they are often large economic ore bodies with long lived mines.

Geologically, porphyry systems consist of stucturally controlled hydrothermally altered rocks which most typically occur at subduction zones on plate boundaries (Sillitoe, 2010). As such, they often occur in clusters, or trends; one of which passes through the study area in British Columbia (see figure 5.4 for a map of porphyries around the world).

![Figure 5.4: Major porphyry systems around the world, taken from John et al., 2010.](image_url)
5.2. The Target: Porphyry Copper-Gold Systems

Targeting porphyry systems is typically done using a combination of geology, geochemistry and geophysics (Berger et al., 2008). Once the correct geological assemblage is identified, geochemistry can be instrumental in identifying the correct alteration zones to vector in on the core of the system. Since porphyry systems are structurally controlled, geophysical techniques such as airborne magnetics and gravity are often able to identify intrusions, faults or contacts, and the alteration zone associated with the deposit often creates a halo effect due to enrichment and depletion of magnetic minerals. Electromagnetic surveys are also effective at delineating the zonation of the porphyry system, and induced-polarization surveys can be particularly useful for mapping the disseminated sulfides characteristic of such deposits.

![Figure 5.5: Illustrative example of different signatures over a copper-gold porphyry deposit (Alumbrera deposit in Argentina), taken from (Hoschke, 2010).](image-url)
5.2. The Target: Porphyry Copper-Gold Systems

![Diagram of a copper porphyry system](image)

Figure 5.6: Diagram of a copper porphyry system, taken from (Sillitoe, 2010).

[Sillitoe, 2010]
5.3 Exploration Goals

The goal in applying mineral prospectivity mapping to the QUEST region is to test the two algorithms (SVM and CNN) and assess their utility as an exploration tool. Though part of the reasoning for turning to machine learning algorithms for such applications is to remove the implicit subjectivity and bias of expert interpretation, it is important to understand that the role of expert domain knowledge in data selection and preparation is of utmost importance. It is also always worthwhile applying common sense when interpreting the results presented from any computer algorithm, and thinking critically about how the results were achieved.

Furthermore, it is important to note that mineral prospectivity mapping (on a regional scale such as performed in this research) does not aim to target ore bodies or drill holes, but rather to highlight regions more likely to bear the target mineralization style. As such, there is no guarantee of success, but rather a diminished risk of exploration in the regions of interest.

5.4 Data Preparation

Publicly available data layers were acquired from GeoscienceBC, the Natural Resources Canada Geophysical Data Repository, and DataBC’s online catalogue. Once these were all downloaded, they were imported into Quantum GIS - an open source geographical information system - to be visually analyzed and prepared.

One of the first steps in preparing the data is deciding which input layers are relevant to the learning problem at hand. A common axiom in machine learning is “garbage in, garbage out”, meaning that despite the desire to allow the data itself to lead the learning process, expert knowledge in the selection of input data is crucial to successful training of the network. For the detection of prospective copper-gold porphyry systems in the QUEST region the geoscientific datasets which were considered
5.4. Data Preparation

of most importance were the following (in no particular order): airborne isostatic residual gravity data, airborne residual total magnetic data, airborne VTEM electromagnetic data, bedrock geology class, bedrock geology age, bedrock geology primary minerals, bedrock geology type, faults, and geochemical pathfinder elements such as Cu, Au, Mo, Ag, As, Sb, Se, U, W, Cd, Ca.

Once all these data layers were loaded into QGIS, some geoscientific processing was performed, such as the following. It is well known that porphyry systems are often structurally controlled, and that such structures can often be imaged using directional filters of the geophysical data, therefore derivative products of the magnetic and gravity data were taken. Additionally, since faults are very important, proximity to a fault system, and the degree of fracturing can be useful metrics. For the categorical geological data such as class (ie: sedimentary, volcanic etc), inputs were split into binary indicator layers so that relationships were not incorrectly inferred from the arbitrary ordering of a discrete representation (ie: volcanic rocks are not more similar to sedimentary rocks than they are to metamorphic rocks; they are simply in alphabetic order). Finally, for regional geochemical data it is often important to account for variability in the background signal since elemental composition can vary greatly over large areas, and can often be attributed to changes in the bedrock, the surface environment, or other factors not directly related to mineral prospectivity.

Since each dataset was collected independently with its own sampling scheme, all layers were resampled to a base grid of 300x300m, resulting in over 700,000 sample points. When all data were assembled and properly processed for training, 91 distinct input layers were prepared, including both continuous and discrete values.

In addition to the training data, training labels were acquired from the BC Minfile database. This database contains a record of all documented mineral occurrences in the province, along with relevant information such as the location, the status of the occurrence (ie: anomaly, showing, prospect or mine) and the mineralization type (ie:
5.4. Data Preparation

alkalic Cu-Au porphyry). Using this information it was possible to create a set of 155 known alkalic Cu-Au porphyry occurrences to use as positive training labels.

Next the data needed to be sampled for the machine learning algorithm, and labels prepared for training. Since the two methods, support vector machines and convolutional neural networks, operate on different inputs (SVM treats point values, CNN uses images), the preparation was specific to the method.

For support vector machines, the resampled grid of 700,000 points on 91 input layers was easily made into a large data matrix. To remove bias from different layers all inputs were normalized and scaled from 0-1. Uncertainty on these values can vary widely depending on the data source. For example, most geophysical data can bear uncertainty in the form of a noise floor plus an acceptable standard deviation, while for geological data it is less obvious due to the subjective, interpreted nature of the measurements. In these cases estimates can still be made based on confidence in the expert and the availability of field measurements. For simplicity during this field test, data uncertainties were not applied.

As previously alluded to, the labels for this application present a suite of practical issues. The lack of confident negative labels (no mineralization) results in an imbalanced learning problem, and the sparse subjective nature of the positive labels (mineralization) results in a large range in confidence which can be adequately quantified using a framework of uncertainty estimates. For the QUEST dataset, the mineral occurrences were used to generate a set of binary labels on the base grid using a radial basis function spline to interpolate between values. Each occurrence has associated with it a status ranging from ‘Showing’ to ‘Producer’ (six unique statuses are possible), indicating the confidence in the mineral occurrence being economic. Combining this with other factors such as the extent of the overburden (which conceals potentially mineral-bearing bedrock), uncertainty estimates for the labels (see figure 5.7) were generated ranging from 1 (confident label) - 50 (not confident label).
5.4. Data Preparation

To feed all of this geoscientific information into a convolutional neural network, the data must be parsed into windows which represent a single patch of the region of interest. Positive training sample locations were cherry-picked using the mineralization locations, and to balance out the problem with negative examples an equal number of locations were chosen from the QUEST region to represent the non-mineralized training labels. These locations were randomly selected from a subset of locations which were greater than 500m from any known mineral occurrences, and should therefore represent the average background (non-mineralized) signal of the QUEST region.

Given this set of positive and negative training image locations and a window size of roughly 10km x 10km, selected based on the expected footprint of a porphyry system, a number of training images were extracted from the dataset. Additionally, to allow for translational and rotational invariance, each window can be shifted and rotated a number of times so as to not bias the training. In other words, for a given patch, the algorithm should detect if there is a porphyry target somewhere in the window; it should not matter where or which orientation. Similar methods can be used to achieve scale invariance in the prediction stages of the process: by scaling the data up and down prior to feeding it to the network, the size of an anomaly will bear much less importance than the structure and pattern of the input data.
5.5 SVM Prospectivity Mapping

5.5.1 Problem Setup

To assess the success of the algorithm, the data was split into training, validating, and predicting sets. Since all of the known mineralization occurs in the north and south of the QUEST region (the central portion is obscured from exploration by a thick overburden), the south was selected for training, the north for validating, and predictions were made for the central region where no mineralization has been discovered to date.

5.5.2 Results

At the request of NEXT Exploration the full result using all data layers is not shown, however the mineral prospectivity map for the validation set (the northern section of QUEST) is shown (Figure 5.8) indicating which regions are more favorable for copper porphyry mineralization than others. As one can see the algorithm was able to successfully predict the known prospective regions, as well as illuminate potential new areas of exploration. The addition of uncertainty estimates in the algorithm provides a more robust framework for the incorporation of multi-disciplinary data which possess a large range in data quality.
5.5. SVM Prospectivity Mapping

Figure 5.7: **Left:** Known mineralization locations in the validation set of the QUEST project area. **Right:** Uncertainty estimates for the mineralization labels (red is more uncertain) in the validation set.

Figure 5.8: Prediction for prospective mineral regions in validation set of the QUEST area (red is more prospective, blue is less) from SVM.

For a more fair comparison with the results from the CNN experiment in the following section, as a demonstration on the full QUEST study area a separate run of the SVM algorithm was completed using only the potential field data (magnetics and gravity). The prospectivity map generated from this experiment (see Figure 5.9) is similar to that using all the data, however it is far less discriminating since less information is being used. When compared to the maps of the individual data layers
5.5. SVM Prospectivity Mapping

(magnetics and gravity) small simpler runs such as this can be instructive as to how the algorithm is combining the information.

Figure 5.9: Prediction for prospective mineral regions in QUEST area (red is more prospective, blue is less) from SVM using only potential field data.
5.6. CNN Prospectivity Mapping

5.6.1 Problem Setup

As a preliminary trial run of real data, the gravity, magnetics and faults were used as inputs for a convolutional neural network bearing a similar architecture to that used for the synthetic example. The first layer had eleven $5 \times 5 \times 3$ convolution kernels, each of which using a sigmoid nonlinearity followed by a $2 \times 2$ softmax pooling function. The second layer had fifteen $5 \times 5 \times 11$ convolution kernels, each of which again using a sigmoid nonlinearity and followed by a $2 \times 2$ softmax pooling function, the third layer had twenty-seven $3 \times 3 \times 15$ convolution kernels, again using a sigmoid nonlinearity and followed by $2 \times 2$ softmax pooling, and the fourth and final layer was a softmax classifier. This resulted in a total of 8702 parameters in the network.

Similar to the SVM example, the data was split into a training and validating set (the north and the south). Since CNNs are nonconvex, and therefore sensitive to initial conditions, 20 runs were attempted with different random initializations, and the best results were reported.

5.6.2 Results

Encouragingly, even this limited training data set, with a relatively simple CNN architecture was able to achieve fairly good results, with the best of 20 randomly initialized runs converging to 80.8% accuracy on the validation set in under 70 iterations (see Figure 5.10 and Table 5.1 for more details). Using the trained classifier to predict the mineral prospectivity for the full QUEST region we were able to create the map seen in Figure 5.11 wherein red indicates the most favourable zones for copper gold porphyry mineralization.
5.6. CNN Prospectivity Mapping

Figure 5.10: Convergence plots for the best result running the QUEST dataset in the CNN. Blue curve is training data, orange curve is validating.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Success %</th>
<th># Correct</th>
<th>False Positive</th>
<th>False Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>74.3</td>
<td>165/222</td>
<td>17</td>
<td>40</td>
</tr>
<tr>
<td>Validating</td>
<td>80.8</td>
<td>42/52</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5.1: Results from classification of QUEST data. Numbers reported are the best achieved using the user-selected CNN architecture over 20 randomly initialized runs.
Figure 5.11: Mineral prospectivity map generated from CNN for QUEST region. Colorscale maps the probability of mineralization from 0-1 (blue-red). Known mineral occurrences are plotted as black dots for reference.
Chapter 6

Conclusions

6.1 Major Contributions

The goal of the research presented in this thesis was to explore, develop and apply advanced machine learning algorithms for mineral prospectivity mapping. This research began with an investigation of current methods for mineral prospectivity mapping which highlighted two important aspects of the problem which to date had not been adequately addressed.

1. The geoscience data available for prospectivity mapping is inherently fraught with highly variable levels of uncertainty, both on the training data as well the training labels.

2. Targeting mineral deposit systems requires a regional approach, focusing on interesting structures/patterns in the data rather than single anomalous values.

To address the first of these difficulties, support vector machines were selected as an algorithmic starting point due to their success and popularity in numerous applications over the last two decades. To incorporate the necessary flexibility to handle uncertainties on both training data and labels the algorithm was modified in two different ways. The first was a simple approach which reweighted the label misfit by a term proportional to the label uncertainty and added an extra term to the objective function which penalized errors in the data using a Gaussian assumption.
6.1. Major Contributions

This new objective function could then be solved iteratively for the model weights \( w \) and the ‘true’ data \( X \).

\[
\Phi(w, X) = \frac{C_1}{2} w^\top w + \frac{1}{\epsilon} \max(0, 1 - \text{diag}(y) (Xw + b)) \\
+ \frac{C_2}{2} (X - X_{\text{obs}})^\top \text{diag} \left( \frac{1}{\Sigma} \right) (X - X_{\text{obs}})
\] (6.1)

Another algorithm was coded beginning from the observation that support vector regression can be rewritten to look very similar to a regularized least squares problem (ridge regression) and that uncertainties in the data could be incorporated by taking a total least squares approach.

\[
\min_{\Sigma, w} \|\text{diag}(\Gamma)(\hat{\Phi}w - y)\|_1 + \frac{\beta}{2} \|\Sigma\|_F^2 + \frac{\alpha}{2} \|w\|^2
\] (6.2)

Following some re-arranging and linear algebra the problem can be broken into a mixed \( L_1/L_2 \) recovery problem again solving iteratively for the model weights \( w \) and the data errors \( \Sigma \) using the following two elegant optimization problems:

\[
\min_w \|\text{diag}(\Gamma)(\hat{\Phi}w - y)\|_1 + \frac{\alpha}{2} \|w\|^2
\] (6.3)

and

\[
\min_v \|\text{diag}(\Gamma)(\gamma \odot v - \hat{y})\|_1 + \frac{\beta}{2} \|\gamma \odot v\|_F^2
\] (6.4)

where \( \Gamma \) and \( \Omega \) are the uncertainty on the labels and the data, respectively, \( \hat{\Phi} = \Phi + \Sigma \), \( \hat{y} = y - \Phi w \) and \( \gamma = \text{diag}(\Omega(w \odot w)) \). The two optimization problems are coupled via the relationship

\[
\Sigma = \Omega^{\odot 1} \odot vw^\top
\]
6.1. Major Contributions

This algorithm, dubbed $L_1$-RSVR (robust support vector regression) was tested on a synthetic data set to illustrate the importance and effectiveness of including the uncertainties on both the data and the labels (when available). This data contained two separable classes of points sampled from respective gaussian distributions, which were confounded by inseparable points having large uncertainty either on their data or their label. The results from this toy example clearly highlight the importance of including uncertainty estimates when the data has variable reliability.

To demonstrate the utility of the algorithm for the mineral prospectivity problem, it was used to generate a prospectivity map for copper-gold porphyry systems in the QUEST region in central British Columbia. The QUEST study contains a large array of publically available geoscience data, including airborne geophysics, multi-element geochemical assaying, and geological mapping of the bedrock. All told, after preprocessing the training data consisted of a matrix of over 700,000 (resampled) point measurements on 91 different data layers. The training labels were taken from the BC Minfile database, using 155 alkalic copper-gold porphryries to construct a binary label for each sample location with an associated uncertainty derived using the information available from the Minfile database in conjunction with the mapped overburden in the region.

To assess the success of the algorithm, the data was split into training and testing sets. In this way it was possible to validate the results on the known mineral occurrences. From these experiments, the support vector machine based algorithms seemed to be performing quite well, however the problem of incorporating spatial information into the learning problem remained a challenge.

To address this second important aspect of the mineral prospectivity mapping problem, we turned to convolutional neural networks. CNNs have become ubiquitous in the fields of computer vision and image identification and segmentation. As computational resources have continued to expand and develop, the power of neural
6.1. Major Contributions

networks has been unlocked, and they have once more gained favour in both research and industrial communities through the resurgence dubbed ‘deep learning’.

Convolutional neural networks owe their success in visual tasks to the observation that natural images contain many localized repeating structures. This hierarchical structure is leveraged by CNNs using multiple layers of convolution to represent complex signals as combinations of simple patterns.

Due to their popularity, many open-source packages exist to implement CNNs (Torch, Theano, Caffe, TensorFlow), however for greater control and to better understand the inner workings of the algorithm we coded our own using the JULIA programming language. This provided greater flexibility and allowed simple implementation of different solvers (including the chosen nonlinear conjugate gradient solver) and modifications such as the inclusion of uncertainties on the labels.

To demonstrate the ability of a CNN at detecting coincident target signals a simple synthetic problem was designed involving three input channels. The first channel is a smoothly decaying ellipsoid with a randomly chosen origin and radii, the second is a smoothly decaying ring, also with randomly chosen origin and radii, and the third is a discrete block with random origin and dimensions. These were chosen to represent simplified versions of geoscientific data where often a high coinciding with a halo structure and the correct geological rock type might be indicative of mineralization (a positive training label, and when the three structures don’t line up this is a negative training label).

Since neural networks are nonlinear and nonconvex, they are sensitive to initial conditions, and therefore 20 runs were completed with different initializations. For the best run, the success rate (on the validation set) was over 90% using a relatively small and simple network architecture. This was deemed convincing evidence that CNNs could perform well on the mineral prospectivity mapping task and therefore it was worthwhile attempting the algorithm on the real data from the QUEST region.
To simplify the first run and reduce computational demands only 3 input channels were used for this: gravity, magnetics and faults. Again 20 runs were completed with different initializations, and the best performing run achieved over 80% success on the validating set.

6.2 SVM vs CNN Comparison

Though the two algorithms used in this research were selected for different purposes, and the goal was never to compare them, it would be natural to question which was better. The support vector machines were chosen due to their elegant formulation and the ease with which one can implement uncertainties. They are also effective at generating a sparse classifier via the selection of support vectors. Because of these factors, it is straight-forward to implement a SVM code which incorporates uncertainty and is able to handle large datasets without prohibitive run times or computational requirements. An additional advantage of support vector machines is the ease with which results can be interpreted and investigated, since the learnt weights essentially highlight which inputs are more or less important for the classifier. Querying a neural network is not nearly so straightforward, and although it is possible to visualize the intermediate feature maps (Zeiler and Fergus, 2014) it is neither simple to implement nor obvious to interpret.

On the other hand, despite the ability to use the kernel trick to achieve non-linear transformations of the data, SVMs can be somewhat limited in their ability to represent the data in an optimal feature space since the kernel must be specified a priori. The strength of convolutional neural networks is their ability to extract high level features from the data via successive layers of convolutional filters. Since these filters are trained during the learning process, they are dynamic and adapt to the data as required. The trade-off for this is that CNNs are much more complex...
models and therefore both more challenging to modify as well as more computationally demanding.

As was discussed previously, one of the main advantages of convolutional neural networks over support vector machines is the ability to recognize anomalous structure in the data rather than simply anomalous values. This can be of great importance in mineral exploration as it is often the structures which are of interest for targeting. It is possible to incorporate spatial information into support vector machines in a rudimentary way via neighbours. One can simply augment the training dataset by adding channels for each of the neighbouring sample locations, resulting in a data matrix of size $n \times md$, where $n$ is the number of sample locations, $m$ is the number of neighbouring points to include (the size of the stencil) and $d$ is the number of data channels (i.e.: magnetics, gravity, geological age, etc). The down-side to this approach is that the size of the data grows rapidly, and to achieve the same spatial sensitivity as the convolutional neural network would require a prohibitively large data matrix (a stencil of $34 \times 34$ pixels).

Comparing the prospectivity maps from the two methods is not totally fair since the support vector machine was trained with the full dataset whereas the convolutional neural network only used 3 (of the available 91) data layers. Despite this, a quick comparison does reveal many similarities, giving greater confidence to both results since each algorithm was run independently.
6.3 Remaining Challenges & Future Work

A natural next step for this research would be to extend the convolutional neural networks to use the entire geoscience dataset. Since structure is treated rather than sample point values, some thought must be given to the handling of categorical variables such as rock class so that boundaries between different units are still recognized. The added input layers would likely require a larger architecture (more neurons, and possibly more layers) to adequately represent the added variability in the inputs.

One of the great challenges with using convolutional neural networks is architecture selection. There exists no clear rule or guide for choosing how many layers to use, nor how many neurons per layer, nor which nonlinearity to apply (though certain types, such as the ReLU (Nair and Hinton, 2010), have become very popular). This remains one of the most frustrating parts of working with neural networks. The architecture selected for this research was chosen somewhat arbitrarily through trial and error, though it is possible alternate choices would perform better.

Since the mineral prospectivity mapping problem is imbalanced and only positive labels are available, assumptions were made constructing the negative labels which may or may not be valid. For the SVM algorithm a spline with radii proportional to the Minfile status (anomaly, prospect, producing mine, etc) was used to interpolate between the copper-gold porphyry mineral occurrences in the database. Uncertainties were used to give very little confidence to all non-mineralized points, as well as data locations known to have thick overburden (since this drastically limits the ability and confidence to prospect and map geology).

For the convolutional neural networks a different approach was taken to generate the labels since the data was windowed into $34 \times 34$ images as inputs. Windows were used centered on each mineralization location as positive labels, and negative labels were randomly selected from a subset of locations which were over 500m from a known
6.3. Remaining Challenges & Future Work

mineral occurrence. This distance was selected through trial and error, since larger
distances removed the majority of interesting regions (the north and south) from the
negative labels, resulting in the majority of the negative training labels being clustered
in the central QUEST area, covered by overburden. The goal in selecting negative
labels in this manner was that the variability in these examples should approximate
the background signal of the various data layers, however it is possible that some of
these locations overlap mineralized zones and therefore are actually positive labels.
Given the relatively small size of the training set, selection of training labels can have
a large impact on the classifier, and therefore both algorithms will likely be sensitive
to these choices.

One possibility for combining the strengths of the two algorithms is to use a con-
volutional neural network with a support vector machine classifier, as in (Nagi et al.,
2012) or (Huang and LeCun, 2006). This architecture would leverage the powerful
feature extraction of the convolutional neural networks with the discriminatory power
of the support vector machines. It can be shown that doing so is equivalent to con-
structing an optimal kernel (using the CNN) for use with the SVM. In this framework
it would also be easy to incorporate uncertainty to the labels.

Adding uncertainty to the data in the convolutional neural network framework
is an open problem which has yet to be examined. Due to the recursive nature of
the architecture and the nonlinear transformations care will be needed to ensure that
both small and large values do not blow up. This remains unexplored and requires
extra thought for proper implementation.

Finally, it is worth noting that as with many applications, greater importance
should be placed on the quality of the dataset than the technique used for analysis.
While algorithm selection will no doubt impact the success of mineral prospectivity
mapping, the availability, quality, and processing of geoscience data will have much
stronger influence on the results. This includes the scale of the data (how many
kilometers are covered; just how regional is the data?), the resolution of the data (how finely is the data sampled; what is the smallest feature to be trusted?), the quality of the data and availability of uncertainty estimates (how precise, and how accurate is the data; is this quantified?), as well as the processing steps used to prepare the data (interpolation, gridding, filtering, etc). For these reasons two important take-away points are worth remembering:

1. Though a powerful analytical tool, mineral prospectivity mapping is not always applicable; using data-driven methods requires that the data is sufficient to guide the learning process!

2. Even in scenarios when mineral prospectivity mapping applies, expert knowledge is crucial in acquiring and processing the data prior to training, and in critically examining results after.
Bibliography


Bibliography


Bibliography


Bibliography


Bibliography

potential mapping: a case study from Aravalli Province, Western India. *Natural resources research*, 12(3).


Bibliography


