Classification and Regression Framework for Characterizing Contaminant Source Zone

A dissertation submitted by

Hao Zhang

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

Tufts University

May 2015

Adviser: Eric Miller

Abstract of "Classification and Regression Framework for Characterizing Contaminant Source Zone"

In this thesis we develop two machine-learning frameworks for estimating quantitative metrics characterizing subsurface zones of chemically contaminated soil focusing on problems involving Dense Non-Aqueous Phase Liquid (DNAPL). Source zone characterization, a necessary first step in the development of the remediation strategy, is challenging due to practical constraints associated with the data available for processing. We first propose a set of geometric features which are based on morphological image processing operations. These features are used for both the classification work in Chapter 3 and the regression approach developed in Chapter 4. Second, we propose a classification framework as our initial solution. Specifically, we quantize each metric into a number of intervals and employ machine learning methods to determine the interval containing the metric. A classification scheme based on an iterative algorithm of Linear Discriminant Analysis (LDA) and Spectral Clustering (SC) is used to determine feature-based clusters that are associated with metric intervals.

Furthermore, we propose a regression framework focusing on the use of manifold regression techniques. We use manifold methods for jointly representing labeled training data comprised of metrics as well as features. We then propose a new integrated approach to the problems of (a) robustly embedding test data into the manifold and (b) constructing a regression function for metrics estimation. The utility of the approach is enhanced by the

explicit incorporation of a physical constraint associated with the metrics into the problem formulation. Results based upon simulated data using Sequential Gaussian Simulation (SGS) method demonstrate the potential effectiveness of the manifold regression approaches as well as significant improvement in performance relative to the case where the algorithmic components are designed serially. At last, we apply our manifold regression algorithms to a new simulated data set whose the hydraulic conductivity fields were built by Transition Probability Markov Chain (TP/MC) model. In TP/MC data the full concentration data are available for training, but the test data are sparsely sampled from 25 ports. The modifications of our manifold regression algorithms to process the sparse data are proposed and the results show the efficacy of our approaches.

Acknowledgements

This work is supported by the Strategic Environmental Research and Development Program Project ER-1612. I would like to greatly thank my thesis advisor Dr. Eric Miller for his support during my Ph.D. program. He inspired and gave me wonderful suggestions for all the achievements in this thesis. I would also thank Dr. Linda Abriola, Itza Mendoza-Sanchez, Shuchin Aeron, Yue Wu and Brian Tracy for their valuable suggestions on this work.

Contents

List of Tables

List of Figures

Classification and Regression Framework for Characterizing Contaminant Source Zone

Chapter 1

Introduction

1.1 Source Zone Characterization

Remediation and restoration of sites contaminated by hazardous Dense Non-Aqueous Phase Liquid (DNAPL) such as trichloroethylene (TCE) and perchloroethylene (PCE) are important problems, primarily because of the persistence of these substances in the subsurface and the danger they pose to drinking water aquifers [66]. A critical component in the planning of a remediation approach and the monitoring of the cleanup effort is characterizing the source zone [38] i.e., the region of the subsurface in which contaminant mass is located. The problem of source zone characterization is complicated by the fact that the distribution of contaminant is determined to a large extent by the spatial variability in hydraulic conductivity, which is typically modeled as a random process whose statistics may be known for a given site but whose specific spatial distribution is certainly not known [19]. After the contaminants are accidentally released, their distribution in the subsurface is determined by the physics of flow and transport through porous media [1]. They are entrapped above

low hydraulic conductivity regions with high saturation or in high hydraulic conductivity regions with low saturation by capillary force.

Christ *et al.*'s paper [22] indicates that a detailed description of the subsurface may be unnecessary for remediation planning. Indeed, the authors in [22] demonstrated that knowledge of a single metric, the ratio of volume occupied by ganglia (regions of contaminant saturation below the residual saturation level S_r) to that of pools (regions above S_r) could be used to predict remediation performance using a simplified model of subsurface flow. The motivation behind the use of *ganglia to pool ratio* (GTP) is that ganglia and pool regions exhibit distinctly different mass flux discharge behavior. In general, ganglia regions produce high concentration contaminant flux signals and are dissolved very quickly; whereas pool regions produce lower concentration but sustained signals.

GTP has been shown to be valuable in studies of simplified models that relate reduction in down-gradient contaminant concentration to the level of mass removal during remediation. In [21], the authors presented a predictive upscaled model where the source zone architecture exponent is a function of GTP. In [22], the authors improved the predictive capability of the model in [21] by separating the contributions from ganglia and pool dominated domains to the dissolved phase, incorporating evolution of pool fraction (f_p) as a surrogate of GTP (f_p is the percentage of the source zone mass incorporated in pools) [1], and accounting for the initial fraction of the concentration eluting from pools region. Therefore source zone architecture metrics (GTP or f_p) and information about distinct ganglia and pools region are key to predict remediation performance. The determination of GTP or

[1]Unfortunately, because GTP can assume values from zero to infinity, we have found it difficult to stably estimate from down-gradient concentration data. On the other hand pool fraction, which is bounded between zero and one, has proven easier to estimate and is easily related to GTP as GTP$= (1 - f_p)/f_p$.

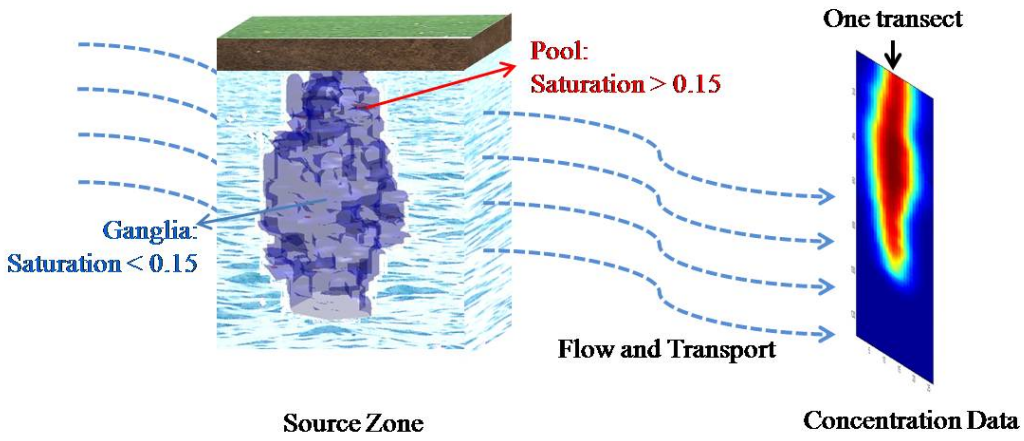

Figure 1.1: The source zone plotted in 3D is modeled as being composed of two parts: "pools" for which the saturation exceeds 0.15 and "ganglia" for which the saturation is lower than 0.15. Flow through the source zone gives the down-gradient concentration data in 2D.

pool fraction, from typically available field data was not considered in [22].

As illustrated in Figure 1.1 and motivated by the ideas in [22] here we address a broader problem of simultaneously estimating a number of metrics describing the source zone based upon observations at a single instant in time of down-gradient DNAPL concentration collected across a transect oriented perpendicular to the flow. Of specific concern to us are pool fraction, f_p; the mass of DNAPL in the source zone occupied by pools, M_p; and the mass of DNAPL in the source zone occupied by ganglia, M_g.

Determination of these metrics is complicated due to the fact that our uncertainty regarding the subsurface encompasses more than the distribution of contaminant. Most notably, the hydraulic conductivity is also typically not known with high precision [22]. In practice, hydrological scientists possess only soft information concerning the statistics of this quantity. Given the availability of high quality computational models for subsurface flow and transport, we consider the use of machine learning methods for metrics determination

[6, 47]. In more details, given a statistical model of the conductivity along with numerical models for both DNAPL entrapment and subsequent flow and transport, the idea here is to simulate a large number of conductivity fields, spill scenarios, and observations of down-gradient concentration from which one can then infer a mapping from concentration data to the metrics.

1.2 Background on approach for processing

1.2.1 Introduction

Based on the discussion in Section 1.1, the problem of interest is the determination of metrics including pool fraction (f_p), mass of DNAPL in pools (M_p) and mass of DNAPL in ganglia (M_g) from observations of a single down-gradient DNAPL concentration image. One could, for example, use an inverse problem approach [4] in which the observed concentration image in conjunction with a flow and transport model is employed in an attempt to estimate both the conductivity and the three-dimensional saturation distribution from which the metrics could then be computed. The severely ill-posed and nonlinear nature of this approach makes it a rather unattractive option and it is impossible to reconstruct the source zone only based on the concentration data sampled at down-gradient [3]. Moreover, as indicated above, we do not seek the full map of saturation but rather are concerned only with determining several quantities, (i.e., f_p, M_p, M_g). Thus a full-blown inversion approach with all of it attendant difficulties is really not warranted.

An alternate approach that does not require full knowledge of subsurface quantities such as hydraulic conductivity is to make use of methods from the field of machine learning [6].

Here it is important to make a distinction between the data that are used to learn this mapping, also known as training data, and testing data that are not associated with the learning process which are used to determine the accuracy of the algorithm. In the training stage, we use the data/metric pairs to learn the mapping, while in the testing stage we use independent data from training data set to evaluate the accuracy of the approach. In this work, both the training and testing data are drawn from simulations. In practice, however, the testing data would come from observations taken in the field.

Additionally, we note that while the training process can be quite computationally intensive, it is performed entirely off-line. The testing procedure, that is, the processing of real data, is quite efficient. For classification approach, as we discuss later in Section 3.3, processing is accomplished using the well known k-nearest-neighbor algorithm [23] which requires only $\mathcal{O}(Nlog(N))$ for searching k-nearest-neighbors, N is the number of training data. For regression approach, the computational complexity is $\mathcal{O}(N)$ for calculating the manifold coordinates of testing data which is discussed in Section 4.2.

Within the machine learning context, the ideal situation would be to determine a regression function that produces a single, point estimate of metric from concentration observation [39]. Our initial investigation into this idea however leads us first to consider a "relaxed" version of the problem. Rather than using regression methods to determine a specific estimate of metric, we quantize each metric (e.g., for pool fraction, the interval between zero and one) into a number of intervals and employ machine learning classification methods to determine the interval containing the metric for a given datum $\mathbf{x}$ by using the concentration

image and metric pairs. Take the estimation of pool fraction as an example.

$$\hat{f}_p = \mathcal{C}(\mathbf{x}) = \begin{cases} \mathcal{C}_1 & \hat{f}_p \in [0, B_1) \\ \mathcal{C}_2 & \hat{f}_p \in [B_1, B_2] \\ \mathcal{C}_3 & \hat{f}_p \in (B_2, 1] \end{cases} \tag{1.1}$$

The function $\mathcal{C}$ is the classifier, it gives one of labels $\mathcal{C}_1, \mathcal{C}_2$ or $\mathcal{C}_3$ to the datum $\mathbf{x}$, each label represents an interval of pool fraction (e.g. the class $\mathcal{C}_1$ means the pool fraction is between 0 and B_1). The B_1 and B_2 are the boundaries between classes.

Building on what we learned from the classification effort, we next turn our attention to the more challenging problem of generating point estimates of the three metrics of interest. That is, take the pool fraction as an example,

$$\hat{f}_p = f(\mathbf{x}) \tag{1.2}$$

where f is the regression function. Here, a far more sophisticated set of tools from the machine learning literature is required to successfully solve this regression problem.

Within the machine learning literature, it is common practice to extract from data a reduced set of *features* to which one applies any of a number of algorithms for classification or regression [70]. In addition to employing well-studied features related to the statistics of the data (or linear transformation of the data), in Section 3.1, we describe a new set of geometric features, the structure of which is driven by the underlying physics of the problem. This feature extraction method is an image processing method regardless of learning procedure and gives more predictive information to estimate the metrics in both

classification and regression framework, therefore the geometric feature extraction method is used throughout.

1.2.2 Classification Approach

As discussed above, the classification task is a relaxed version of our problem, however it still brings with it a number of challenges that we address in this work. In Figure 1.2 we illustrate the manner in which the feature vectors extracted from training and testing data interact to determine pool fraction. For ease of visualization, we assume that the feature set is two-dimensional. The "+"'s indicate the feature vectors associated with the training data while the " ∘ " is the feature vector extracted from a testing data set. Associated with each class is an interval of pool fraction. Our approach to data processing amounts to looking at all the "+"'s in a neighborhood of the " ∘ " and choosing that pool fraction interval associated with the majority of the neighbors; nothing more than the well-known k-nearest-neighbor method for classification [23].

The difficulty here is that *a priori* the number of intervals and their boundaries are not known. Ideally, the features would naturally cluster into groups associated with disjoint intervals as shown in Figure 1.2, in which case interval determination would be trivial. Reality is a good deal less ordered however. While "high" and "low" pool fraction features do tend to cluster, outside of the extremes, clearly defined grouping are absent. Thus, we have here a clustering problem in which we seek to develop clusters in feature space that ultimately provide high accuracy in terms of metrics classification.

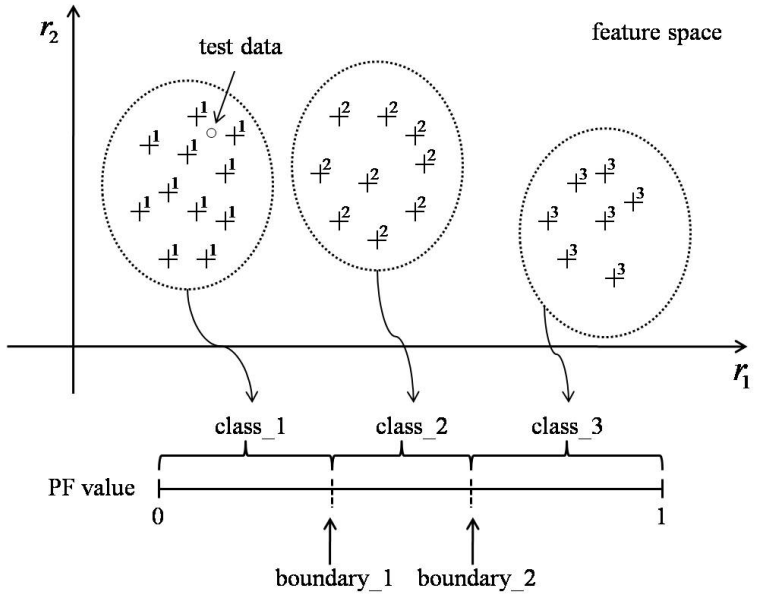

Figure 1.2: An illustration of classification using k-nearest-neighbor algorithm. We illustrate the feature space in two dimensions $(\mathbf{r}_1, \mathbf{r}_2)$,"+"'s indicate the training data and " ∘ " indicates the test data. The number on the up-right corner of each "+" shows the class label, which is corresponding to an interval of f_p value.

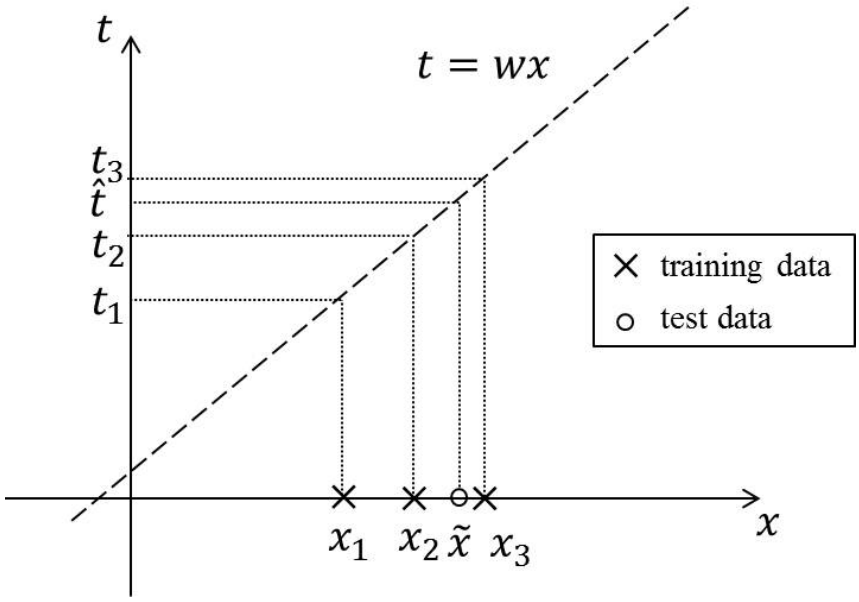

Figure 1.3: An illustration of closeness condition for linear regression function.

1.2.3 Regression Approach

In the regression framework, we utilize a Bayesian method for training the regression functions because it not only gives the estimate of each test datum, but also an associated confidence interval.

For purposes of estimation, we need to transform the feature vectors into a space such that the distance between vectors in this new space is reflective of the distance between the corresponding source zone metrics we seek to determine. If this condition is satisfied, when the feature vector from a test data set is transformed into this space, the use of regression for the metric based on the training data points close to the test data point is expected to be accurate. As illustrated in Figure 1.3, we show the linear regression function $t = wx$ in one dimension, the x is the one dimensional feature of concentration data, the t is the metric we want to estimate and w is the weight vector of regression function. The "$\times$" and "$\circ$" represent the training datum and test datum respectively. Since the test datum $\tilde{x}$ is in the neighborhood of x_1, x_2 and x_3, the estimation $\hat{t}$ of test datum is similar to t_1, t_2 and t_3. This requires the distribution of data in the feature space satisfies the *closeness condition* (known more formally as a locality preserving property [9]), that is, raw data as well as feature vectors which are close in a typical linear Euclidean sense have corresponding metrics that are also close while data/feature vectors that are quite dissimilar correspond to metrics that are different. To develop a deep sense for the ambiguity of the regression problem, it is important to note that the closeness condition does not always hold in the linear Euclidean space of the raw concentration image data where the Frobenius norm is used to measure the distance between the concentration images, i.e., $\|\mathbf{c}_i - \mathbf{c}_j\|_F^2$, or even

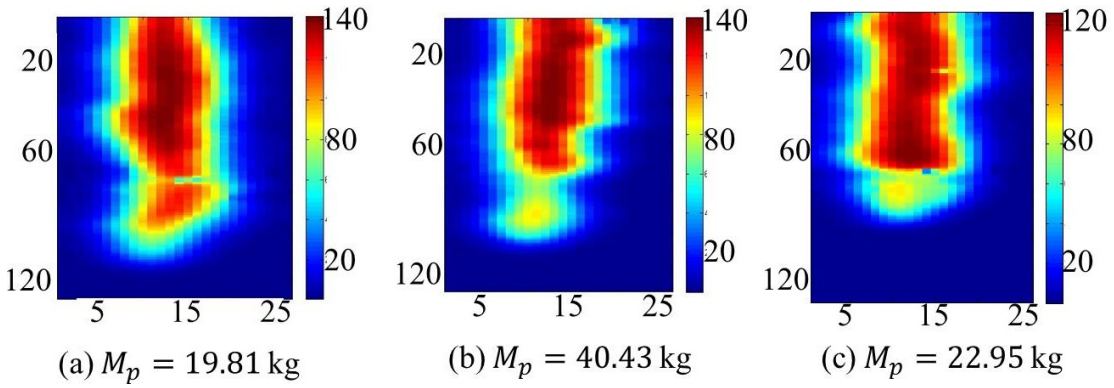

Figure 1.4: Ambiguity of the metric estimation problem. Due to the differences in the manner in which the contaminant was spilled into the subsurface, the pool masses associated with similar concentration images (a) and (b) are quite different while the dissimilar images (a) and (c) correspond to source zones with nearly the same mass in pools.

that of our geometric feature vectors proposed in Section 3.1 where the distance is given by $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$. That is, raw data as well as feature vectors which are close in a typical Euclidean sense may not have corresponding metrics that are also close while data/feature vectors that are quite dissimilar may well correspond to metrics that are close. Therefore we employ manifold dimension reduction method to find a manifold space where the data with similar metrics will be located near-by.

As an example, consider the three concentration images and corresponding pool mass metrics illustrated in Figure 1.4. In Table 1.1, the Euclidean distances between all three pairs of image data and feature vectors are provided along with the distances computed using the manifold ideas developed later in Section 4.1. While cases (a) and (c) have the most similar pool masses 19.81 kg and 22.95 kg respectively, the raw concentration images and the raw feature vectors would predict that (a) and (b) were most similar. Indeed, in manifold space (a) and (c) are placed closest together. Moreover, pairs (a)/(b) and (b)/(c) which have the large mass difference are also placed a large distance apart in manifold space.

	Magnitude of Mass Difference (kg)	$\|\mathbf{c}_i - \mathbf{c}_j\|_F^2$	$\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$	$\|\mathbf{r}_i - \mathbf{r}_j\|_2^2$
Comparing (a) to (c)	3.14	1046	7.43	**0.196**
Comparing (a) to (b)	20.62	**862**	**5.29**	1.05
Comparing (b) to (c)	17.48	912	7.49	0.868

Table 1.1: The distance between the data corresponding to Figure 1.4 showing the ambiguity in our data sets. The pool mass of datum (a) and (c) are almost the same, but the Euclidian measurements of concentration image pixels and geometric feature vectors are large. After nonlinear dimension reduction, (a) and (c) will be located near-by in the manifold space.

The manifold methods give the coordinates of training data directly given the feature/metric pairs, after which the regression functions are trained in this manifold space. Therefore, in order to estimate the metrics for test data, we need to embed them (which obviously will not include the associated metric values) in the same space as the training data. A known challenge of these manifold methods is that lack of an explicit embedding function for the processing of test data where the source zone metrics are not known [10]. As discussed in Section 1.3, we need to learn an embedding function; that is, a mapping from geometric feature space to the manifold space, shown in Figure 1.7. We employ the Spectral Regression (SR) method [12] to learn this embedding function, after which we can embed the test data in the manifold where regression can then be performed.

In our work with SR method, we must contend with a different type of ambiguity from the one motivating the use of manifold methods. Specifically, in some cases, data sets corresponding to similar metrics can be quite different leading to large errors in the embedding. To see this, consider the three concentration data in Figure 1.5 all with M_p values around 1.03 kg. In cases (b) and (c) the peak concentration value is less than 15 mg/L, and the areas where signal is present are small and well localized. These are typical for M_p in this range. Case (a) however is an outlier. The peak concentration about 100

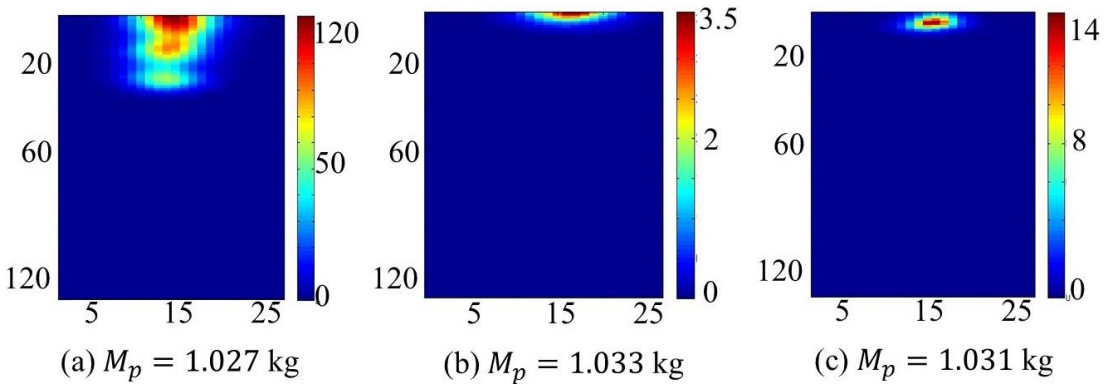

Figure 1.5: The outliers in data set, (a)-(c) have almost the same metric mass in pools, but due to the different spill scenario, the concentration data is quite different. The peak concentration values in (a) are more than 100 mg/L, but the concentration values in (b) and (c) are less than 15 mg/L

mg/L is quite high and the morphology of the concentration image is quite different from that of cases (b) and (c). The reason for these differences can be related to the specifics of the spill scenarios that gave rise to the data. For case (a), the source zone was highly dominated by ganglia that dissolved quite quickly yielding large concentration values in the down-gradient transect even as the up-gradient source zone contains little mass in pools. In cases (b) and (c) by contrast more DNAPL was entrapped above the low hydraulic conductivity region to form pools, when the ganglia were flushed out, there is only pools in the source zone, which gave little concentration in down-gradient transect. In order to decrease the sensitivity of embedding function to outliers, the robust SR is proposed and discussed in detail in Section 4.4.

1.3 Overview and contribution

Given a collection of down-gradient concentration data and metrics from the source zone, we have developed machine learning algorithms to determine a mapping from data to metric

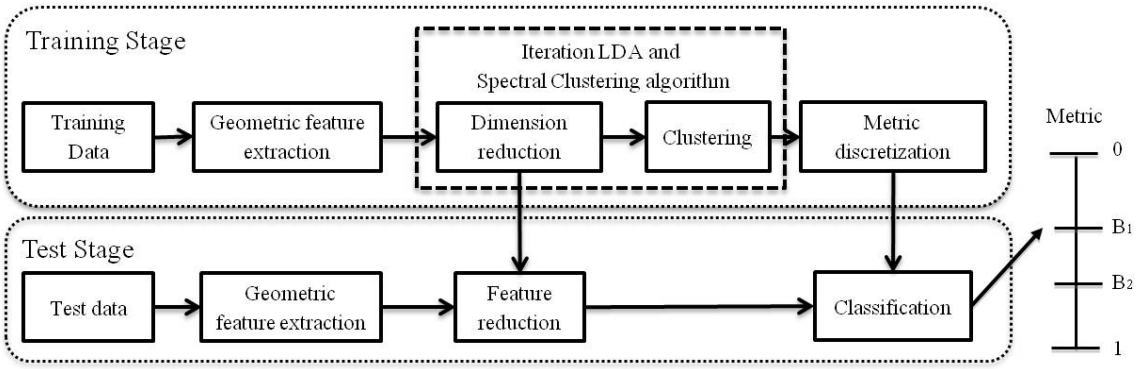

Figure 1.6: The framework of our classification-based machine learning approach. The geometric feature extraction is shared by both training and test stage, because it is an image processing method regardless of training procedure.

such that when a new datum is made available, we are able to predict the associated metrics. In classification framework, the classifier will give an interval of metric where the new datum belongs to; in regression framework, the regression function will estimate one single number for each metric.

1.3.1 Classification Framework

From the concentration data, the first two steps of training stage in Figure 1.6 are feature extraction and dimensionality reduction. These processes are employed to obtain from the raw data quantities that are, in some sense, more predictive of the metric than raw concentration observations. In a bit more detail, learning a good classifier requires that the size of the training set, N, be at least an order of magnitude larger than the dimensionality of the variables we use for prediction. As discussed in Section 2.6.1, for our problem, we have about $N = 500$ concentration images as the basis for training. Hence, we require at most a few tens of quantities for estimating a source zone metric. As described in Section 3.1, the geometric features that we have developed are based on the use of morphological signal

processing methods. Ultimately, the number of such features is still over three hundreds. Dimensionality reduction is used to further extract from this set those degrees of freedom that are most relevant for solving the classification problem. We call these quantities *reduced features*. We make use of an existing classification scheme based on a combination of Linear Discriminant Analysis (LDA) [43] and Spectral Clustering (SC) [17] as the basis for our approach to determine feature-based clusters that are associated with metric bins.

The problem here is that the LDA algorithm assumes that labeled data are available; i.e. the bins into which the metric will be divided are known as a priori. As this is not the case for our problem, we have developed an iterative method that both finds the reduced dimension feature space and cluster the feature vector in this space.

The final two steps in training are clustering and discretization of metric which form the basis for the classification algorithm that will be used to process data from outside the training set. The clustering problem we have here seeks to partition the collection of reduced features in a manner such that those that are close in reduced feature space correspond to source zone metrics that are also close to one another. Then we discretized the continuous metric into several bins according to the clusters in the reduced dimension feature space.

In the testing stage, we use independent concentration data from training to evaluate the performance of classifier. Testing is a two-step process. First, the new concentration datum is transformed into a reduced feature vector. Second, we look for the collection of k points in reduced dimension feature space that are closest to test data. Based on this collection we classify the testing data into one of the metric intervals.

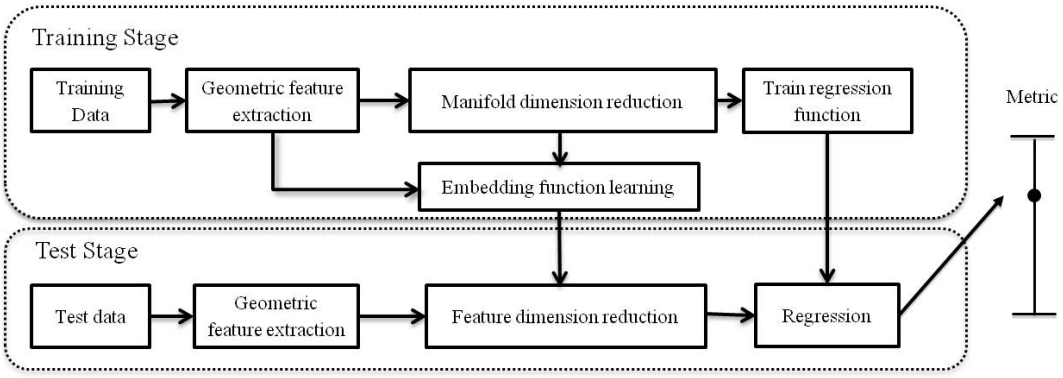

Figure 1.7: The framework of our regression-based machine learning approach. The geometric feature extraction is shared by both training and test stages, because it is an image processing method only applying to concentration images regardless of metrics information.

1.3.2 Regression Framework

The regression framework proposed in this work is designed to learn in a supervised manner (i.e., given labeled training data generated by the models) three nonlinear Bayesian regression functions to estimate the metrics of interest. The idea we propose is based upon machine learning methods that have gathered much attention recently: manifold dimension reduction for obtaining low dimensional representations of high dimensional data and Spectral Regression (SR) for embedding incomplete test data into the manifold.

Motivated by the ideas in [30], our work focuses on the use of state-of-the-art ideas in machine learning to construct regression functions for estimating the three metrics described in the Section 1.1 (pool mass, ganglia mass, and fraction of mass in pools) from the down-gradient observations. This process is comprised of a number of component steps that are illustrated in Figure 1.7 and described in detail in Chapter 4. While the computational models of entrapment and dissolution are certainly of use in generating data for building this regression function, it is still the case that the computational burden of running these models limits the size of the data sets available to us. More precisely, for our problem, the

size of a concentration image is about 3000 pixels while, N, the total number of such images and associated known metrics in our all training data sets, is only about 2000. We must first reduce the dimensionality of the data after which a suitable regression function can be constructed.

Here we follow a two-step approach to reduce the dimensionality: feature extraction from the concentration images followed by a low dimensional, manifold-based representation of this feature set. The geometric feature sets are described in Section 3.1. As the dimension of the resulting feature vectors is still well over 300, we employ manifold learning methods to obtain a low dimensional manifold coordinate vector (the dimension will be four) for training a regression function under a Bayesian approach.

Our use of manifold methods is motivated by more than just computational considerations. The use of this machinery allows us to transform the feature vectors into a space such that the distance between vectors in this new space is reflective of the distance between the corresponding source zone metrics we seek to determine. If this condition is satisfied, when the feature vector from a test data set is transformed into this space, the use of regression for the metric based on the training data points close to the test data point in manifold space is expected to be accurate.

A key challenge in the use of manifold methods for machine learning is the embedding of test data. One method developed in the context of unsupervised problems is the Nyström approach introduced in [10]. Recently, a review of manifold-learning-based feature extraction methods was provided in [46]. In [45] the authors adapted a force field intuition to develop a nonlinear graph embedding framework after using spatial and spectral information to compute the neighborhood graph. In [77] the authors first projected the hyper-spectral

image into a low-dimensional space by minimum noise fraction, then built the dictionaries for sparse coding the geometric features by using wavelet methods. For the supervised case of interest in this work where we seek to embed test data (where labels are not known) into a manifold constructed using knowledge of both the data and the labels, we built on the Spectral Regression (SR) technique [12]. SR was designed to learn an embedding function from any ambient space into a previously constructed manifold. Of particular interest here is the case where the ambient space is comprised of the label-free observations.

After the test data are embedded within the manifold, Bayesian regression functions are used to generate estimates of the source zone metrics. As explained in greater depth in Section 1.2.3, the idea here is to take an estimate of the metric as an optimal linear combination of the metrics associated with the training data embedded in the vicinity of the test data within the manifold.

In summary, the goal of the manifold regression framework in the work is to construct regression functions from a collection of concentration data and metrics pairs $(\mathbf{c}_1, \mathbf{t}_1), (\mathbf{c}_2, \mathbf{t}_2), \cdots, (\mathbf{c}_N, \mathbf{t}_N)$, where $\mathbf{c}_i, i = 1, \cdots, N$ denotes the concentration data and $\mathbf{t}_i$ is the vector of three metrics $\mathbf{t}_i = [f_p, M_p, M_g]_i$, and then given a new data $\mathbf{c}$, we can estimate the metrics vector $\hat{\mathbf{t}}$.

1.3.3 Contribution

The first contribution of our work is that we propose a set of geometric features, the structure of which is motivated by the underlying physics of our problem. Through the observations we find the shape and size of "blobs" in concentration image are closed related to the estimation of metrics. We view the concentration images as the height maps and use a

threshold operation to calculate the number of connected component and the percentage of remaining area above the threshold. In Section 3.1, we show the geometric feature vector can indicate the changes of metrics in source zone. Because the feature extraction method is a kind of image processing methods not involving any learning procedure, it can be used in both classification and regression framework.

In the classification framework, one contribution is an iterative algorithm to reduce the dimensionality of feature vector by using Linear Discriminant Analysis (LDA) and to cluster in the reduced feature space by Spectral Clustering (SC). The algorithm we describe in Section 3.3 is, to the best of our knowledge, unique in machine learning applications. SC method clusters the data located near-by into the same class, thus can give the initial label to each data. Then using the data and the initial label pairs, LDA can find a reduced dimensional space in which the distance between the data in the same class is as small as possible, whereas the distance between each class is as large as possible. In this reduced dimension space, SC updates the label of each data. This iterative algorithm will keep updating the label information until the label of each data stays the same.

In the regression framework, one new contribution in this work is the development of an integrated, variational formulation for the simultaneous determination of the Spectral Regression (SR) embedding function and the associated Bayesian regression function. To the best of our knowledge such an approach has not been considered in general or more specifically within the context of geophysical applications. While one could certainly design these two components separately, for the application driving the work in this thesis we have found that jointly determining the embedding function (which determines the coordinates of the test data in the manifold) and the regression function (which is driven by the coordinates

of the training data in the neighborhood of the embedded test datum) can lead to substantial improvement gains.

In constructing the variational problem defining the embedding and regression functions, we have included a number of elements that we again believe are new. First, within the SR context, only square error norms have been considered [12] for determining the embedding function. As we demonstrate in Section 1.2.3, ambiguities associated with the data available for application cause outlier problems in which test data may occasionally be embedded within the manifold far from appropriate training data. To reduce the impact of these outliers we develop a robust form of spectral regression.

A second novel component of the variational formulation is the use of a physical constraint to tie together the Bayesian regression functions for the three metrics of interest. Specifically, pool fraction is the ratio of pool mass to the sum of pool mass and ganglia mass. The results in Section 4.6.3 indicate that in cases when training data are scarce, enforcing the physical relationship among the three quantities allows for close to the same level of performance as in the data rich case. This is important since it means we have a method for doing as well with less data which may be expensive to collect or time consuming to simulate as we do when data are plentiful.

In summary, the work in this thesis contributes to the state of the art in a number of areas. From an applications perspective, the machine learning methods of both classification and manifold regression have not been considered to date for addressing problems of contaminant source zone characterization. Here we demonstrate that these approaches provide viable techniques for this problem. We have extended the machine learning methods for geophysical applications in a number of ways. For classification, we proposed an

iterative algorithm to reduce the dimensionality of feature vector and cluster in the reduced feature space. For manifold learning techniques, first, we incorporate the label information directly in the objective function of Laplacian Eigenmaps, from which a single low dimensional manifold is constructed for the determination of the three metrics of interest. Second, by using the Huber norm rather than the $L2$ norm we obtain a robust formulation of Spectral Regression. Third, we consider the specific physical constraint between three regression tasks and force this constraint to regularize the manifold regression functions for the three metrics. The experiments in Section 4.6.3 indicate precisely how these elements of our algorithms combine to provide enhanced performance relative to the case where one first constructs a manifold and then separately builds spectral regression functions for embedding and Bayesian regression functions for metrics estimation.

1.4 Outline of thesis

The remainder of the thesis is organized as follows: In Chapter 2 we briefly summarize the hydrological background and review the machine learning materials for classification, regression, manifold learning, out-of-sample extension for manifold learning and the data sets we used for training and testing the performance of our approaches. In Chapter 3, we discuss in detail the classification framework and initial experimental results on a small data set. In Chapter 4 we propose the manifold regression framework upon which our work is based including Laplacian Eigenmaps, Spectral Regression, and Bayesian regression, the experiments evaluate the efficacy of our regression framework. Then our robust version of Spectral Regression is proposed, the third subject of this chapter is an integrated approach

of embedding function learning and regression function learning, the experiments in this chapter demonstrate the superior performance of our integrated approach. In Chapter 5, we applied our manifold regression approaches to process the sparse data generated by 2D model, experimental results of the methods are presented and analyzed. Finally, conclusions and future work are the topics of Chapter 6.

Chapter 2

Background

2.1 Hydrological Model: Multi-phase Flow and Transport Model

As discussed in the Section 1.1, the application motivating the technical developments in this work is the characterization of subsurface zones contaminated by DNAPL such as TCE and PCE used in a variety of economic activities from dry-cleaning to industrial degreasing [81]. Once released in the subsurface, DNAPL tends to distribute themselves in one of two ways: either as regions of relatively high saturation[1] [51] known as pools that form above the areas of relatively low hydraulic conductivity or as more diffuse, lower saturation ganglia. Following convention [22], in this work pools are taken to be connected regions in the source zone where the DNAPL saturation exceeds 0.15 [41] while ganglia are those areas where the saturation is below this threshold.

The spatial distribution of DNAPL saturation is governed by the physics of multi-phase

[1] Saturation of DNAPL is the volume fraction of the total void volume occupied by DNAPL

flow and transport through porous media. In this work, the system of interest contains two fluid phases: DNAPL and water. The DNAPL phase in general is immobile and dissolved when the aqueous phase flows through it. The saturation of each phase (indexed by p) is described by the following partial differential equation presented in Abriola *et, al.*'s work [1],

$$\frac{\partial(\rho_p \varphi S_p)}{\partial t} - \nabla \cdot \left(\rho_p \frac{\mathbf{k} k_{rp}}{\mu_p}(\nabla P_p - \rho_p \mathbf{g})\right) = \sum_{p'} M_{pp'} \tag{2.1}$$

where ρ_p is the density of phase-p, φ is the porosity, S_p is the saturation, $\mathbf{k}$ is the intrinsic hydraulic conductivity tensor, k_{rp} is the relative hydraulic conductivity, μ_p is the dynamic viscosity, P_p is the thermodynamic pressure, $\mathbf{g}$ is the gravity vector, and $M_{pp'}$ is the mass transfer to phase-p within the contiguous phase p' which takes place through dissolution or absorption. In our application, we only consider two fluid phases in source zone saturation, they are DNAPL phase (S_n) and aqueous phase (S_a), which satisfy the following constraint,

$$S_n = 1 - S_a \tag{2.2}$$

The nominal flow of groundwater through the source zone will gradually dissolve DNAPL and create a diffuse plume of aqueous phase contaminant distant from the region containing the DNAPL.

In practice, contaminant site characterization is accomplished through the processing of observations of such down-gradient concentration signals. In this work we consider the case where we have concentration data sampled densely in space across a transect oriented orthogonal to the direction of groundwater flow at a single point in time. While there would be a significant benefit to having data from multiple points in time, the time scales

associated with relevant changes to the concentration data caused by groundwater flow are far too long for most practical circumstances. Additionally, although in the field these transects are constructed from a small number of wells, we initially assume here that a dense collection of data are acquired resulting in the availability of a concentration "image" for processing. Though admittedly an idealization, this assumption allows us to more readily develop and demonstrate the utility of a set of machine learning tools to address the rather challenging problem of source zone characterization from a single temporal snapshot of data.

The concentration value in phase-p (c_p) is computed by the following mass balance partial differential equation,

$$\varphi \frac{\partial (S_p c_p)}{\partial t} + \varphi \nabla \cdot S_p (c_p \mathbf{V}_p - D_p \cdot \nabla c_p) = \sum_{p'} M_{pp'} \tag{2.3}$$

where D_p is the hydrodynamic dispersion tensor in phase-p [8] and $\mathbf{V}_p$ is the pore velocity vector of phase-p determined by the following Darcy's law [1],

$$\varphi S_p \mathbf{V}_p = -\frac{\mathbf{k} k_{rp}}{\mu_p} (\nabla P_p - \rho_p \mathbf{g}) \tag{2.4}$$

the right side of (2.3) is the mass transfer from phase-p' to phase-p which is determined by the following linear driving force expression [74],

$$M_{pp'} = \kappa_{pp'} (c_{p_{eq}} - c_p) \tag{2.5}$$

where $c_{p_{eq}}$ is the concentration value in phase-p in equilibrium with phase-p', $\kappa_{pp'}$ is the lumped mass transfer coefficient which represents the rate of mass transfer of from phase-p'

to phase-p [57].

Here DNAPL is considered to be comprised of only one constitute (i.e., PCE or TCE) and the dissolution is the only mass transfer process between the phases, therefore a simple form of the DNAPL constitute balance is as follow,

$$\varphi \frac{\partial(S_n \rho_n)}{\partial t} = \kappa_{an}(c_{a_{eq}} - c_a) \tag{2.6}$$

Pool fraction (f_p), the metric of interest in source zone remediation, is defined as the percentage of DNAPL as pools:

$$f_p = \frac{M_p}{M_p + M_g} = \frac{\int \rho_n S_n \phi dx dy dz \quad \forall S_n \geq 0.15}{\int \rho_n S_n \phi dx dy dz} \tag{2.7}$$

where ρ_n is the DNAPL density, S_n is the saturation, ϕ is the porosity, dx, dy and dz are for x, y, and z directions respectively. The term in the numerator of (2.7) is the mass of DNAPL in pools (M_p) and the denominator is the total mass ($M_p + M_g$) of DNAPL in the source zone. Since these M_p and M_g are closely related to the f_p, we also expect we can estimate them and hopefully the performance of our algorithm can be improve by the estimation of these three metrics simultaneously. The mass of DNAPL in pools is defined as,

$$M_p = \int \rho_n S_n \phi dx dy dz \quad \forall S_n \geq 0.15 \tag{2.8}$$

and the mass of DNAPL in ganglia is defined as,

$$M_g = \int \rho_n S_n \phi dx dy dz \quad \forall S_n < 0.15 \tag{2.9}$$

2.2 Classification

The goal of the classification framework for our problem is to take a concentration image $\mathbf{c}$ as input and to assign it to one of K discrete classes $\mathcal{C}_k$ where $k = 1, \cdots, K$. Each class represents an interval of the associated metric. Many classification machine learning methods have been used in geophysical image classification applications [31, 14, 15, 52, 13]. There are three kinds of classification techniques depending on the availability of label information by which we mean knowledge of the ground truth classes to which each datum belongs. The first one is called supervised classification, in which the sufficient label information is available and used in the training procedure. The second class of methods is called unsupervised classification, in which there is no prior label information. Rather, clustering-based algorithms are applied to partition the data into clusters based on the features inherent in these data. The third kind of classification techniques is called semi-supervised classification, in which a small set of data has labels and a wealth of unlabeled data are also available. In the learning procedure an initial classifier is trained based on the labeled data pairs, after which the unlabeled data will be classified, and then these new labeled data will be added in the classifier training procedure. For our problem, although the metric information is available for each datum, the number of classes and the boundaries between each intervals are unknown. Thus we will first use unsupervised classification method to find the label for each datum, then use supervised approaches to reduce the dimension of data. Therefore our classification approach belongs to the semi-supervised classification.

In this section, we review several kinds of unsupervised and supervised classification

techniques used in geophysical data processing. Unsupervised classification methods can be used to find hidden patterns in the data with absence of label information [26]. Two cornerstones of unsupervised classification methods are dimension reduction and clustering. One widely applied unsupervised dimension reduction method is Factor Analysis, which assumes the data are generated by a linear combination of latent factors plus noise [60]. In practice, the number of latent factors is much smaller than the dimension of observed data, thus factor analysis can reveal the intrinsic dimensionality of data set by finding the key factors.

Another popular dimension reduction method is Principal Component Analysis (PCA) [67]. PCA is an important limited case of factor analysis which assumes the variance of the noise in the data is zero. A further constraint of PCA is that the factors of data set need to be orthogonal, which makes PCA attractive because the solution is the eigen-decomposition of the covariance matrix of data set [59].

For clustering, the Gaussian mixture model assumes the distribution of data is a linear combination of Gaussian, each of which has its own mean and variance. The density of each data point follows $p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$, where $\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$ is one of K components of the mixture model and π_k is the proportion of each component, thus $\sum_{k=1}^{K} \pi_k = 1$ [58]. Another closely related method to mixture Gaussian model is K-means, the value of density assumes the variance matrix of each Gaussian distribution is identity matrix and $\pi_k = \frac{1}{K}$ [72]. The details about mathematical models of both PCA and K-means are introduced in Section 3.2.

The most well-known supervised linear dimension reduction method for classification is Linear Discriminant Analysis (LDA) [24], which is a well-established method for finding

a linear transformation that, in a precise mathematical sense, projects high dimensional feature vectors onto a low dimensional space in a manner that maximally separates classes, in our case different bins of f_p, M_p, M_g. The mathematic model of LDA is discussed in Section 3.3. The K-means method measures the similarity between data using Euclidean distance and can not guarantee to balance the data in different clusters. Spectral Clustering partitions the training feature sets into several clusters by spectrum analysis of Laplacian graph and is equivalent to normalized cut problem [71] which can distribute the data in different cluster more evenly. The detail of Spectral Clustering is introduced in Section 3.3.

2.3 Regression

Regression techniques can give point estimates of the source zone metrics for each concentration observation, which is different from the classification solution. Regression is employed for problems where the quantity to be determined is continuously varied while classification is used for problems where there are only a finite number of values that the variable can assume. In machine learning, a simple model called multiple linear regression [5] learns a linear function to predict a target variable given one or more observations. In order to avoid *overfitting*, which means the predictors are redundant and the regression model is too complicated, the linear regression model with regularization term called ridge regression model was introduced by Hoerl *et al.*'s work [34]. A nonlinear regression method called artificial neural networks was inspired by animal's nervous system which typically consists of interconnected "neurons" [79]. The input neurons are activated by the data, and then these activations are weighted and transformed through the network, finally the estimation

is given by the output neuron. The model of regression function is determined by the weight factors connecting the neurons and the structure of network. The famous Back-Propagation (BP) algorithm is used to determine the weights [40] adaptively. However the structure of network needs to be designed by human before the training of weights and BP algorithm can't guarantee to find the global optimal solution.

Another popular regression model is ε-Support Vector Regression (ε-SVR), the goal of this model is to find a function $\hat{y} = f(x)$ that the deviation of $\hat{y}$ from the actual target y is at most ε [65]. A soft margin version of ε-SVR allows some error greater than ε by introducing the slack variables. The optimization problem of ε-SVR can be converted to quadratic programming using Lagrange multipliers.

Bayesian regression is a kind of statistical machine learning method, which uses Bayes Law to determine a regression function [11]. Moreover, under a Bayesian framework, rather than a point estimate for the metrics, we provide a full probabilistic model; i.e., a joint density function of the metric estimates given the data. In our case, we assumed it is Gaussian. The details of Bayesian regression is discussed in the following.

We assume the model of linear regression function for training data is $t(\mathbf{r}) = \mathbf{w}^T\mathbf{r} + \epsilon$, ϵ is additive Gaussian noise distributed as $\epsilon \sim \mathcal{N}(0, \beta)$ where for our problem β is determined according to a procedure discussed in Section 4.3. Under this model we have $\Pr(t(\mathbf{r}_i)|\mathbf{w}, \mathbf{r}_i) = \mathcal{N}(\mathbf{w}^T\mathbf{r}_i, \beta)$. Under the standard assumption that the $\mathbf{r}_i$'s are independent, the overall likelihood function is,

$$\Pr(\mathbf{t}|\mathbf{w}, \mathbf{R}) = \prod_{i=1}^{N} \Pr(t(\mathbf{r}_i)|\mathbf{w}, \mathbf{r}_i),, \text{ where } \mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N]. \tag{2.10}$$

According to Bayesian rule, the posterior distribution of $\mathbf{w}$ is,

$$\Pr(\mathbf{w}|\mathbf{t},\mathbf{R}) = \frac{\Pr(\mathbf{t}|\mathbf{w},\mathbf{R})\Pr(\mathbf{w})}{\Pr(\mathbf{t})} \tag{2.11}$$

Assuming the conjugate prior distribution $\Pr(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \alpha\mathbf{I})$, the maximizing posterior distribution is [11],

$$\Pr(\mathbf{w}|\mathbf{t},\mathbf{R}) = \mathcal{N}(\beta^{-1}\mathbf{S}_N\mathbf{R}\mathbf{t}, \mathbf{S}_N), \text{ where } \mathbf{S}_N = (\alpha^{-1}\mathbf{I} + \beta^{-1}\mathbf{R}\mathbf{R}^T)^{-1} \tag{2.12}$$

resulting in the maximum posterior estimate of $\mathbf{w}$ as $\mathbf{w}^* = \beta^{-1}\mathbf{S}_N\mathbf{R}\mathbf{t}$.

The estimate of the metric for a test datum is obtained by the predictive distribution [11],

$$\Pr(\hat{t}|\mathbf{R},\mathbf{t},\tilde{\mathbf{r}}) = \int \Pr(\hat{t}|\mathbf{w},\tilde{\mathbf{r}})\Pr(\mathbf{w}|\mathbf{t},\mathbf{R})d\mathbf{w} \tag{2.13}$$

where $\tilde{\mathbf{r}}$ is the manifold coordinates of test datum in manifold space, $\hat{t}$ is the metric corresponding to test data. Assume $\Pr(\hat{t}|\mathbf{w},\tilde{\mathbf{r}}) = \mathcal{N}(\mathbf{w}^T\tilde{\mathbf{r}}, \beta_{\tilde{r}})$ with $\beta_{\tilde{r}}$ determined as discussed in Section 4.3, the distribution of $\hat{t}$ is,

$$\Pr(\hat{t}|\mathbf{R},\mathbf{t},\tilde{\mathbf{r}}) = \mathcal{N}(\mu, \hat{s}^2) = \mathcal{N}((\beta^{-1}\mathbf{S}_N\mathbf{R}\mathbf{t})^T\tilde{\mathbf{r}}, \beta_{\tilde{r}} + \tilde{\mathbf{r}}^T\mathbf{S}_N\tilde{\mathbf{r}}) \tag{2.14}$$

We take the estimation of $\hat{t}$ as μ and the 85% of confidence interval as $\pm 1.44\hat{s}$.

2.4 Manifold Learning

Within the geophysics community, a wide range of manifold dimension reduction methods have been proposed to address a variety of issues specifically in the areas of geophysical image classification and regression [47, 18, 27, 36, 68, 69, 78]. For example, in Chen *et al.*'s work [18], two manifold dimension reduction methods, ISOMAP and multidimensional scaling, are evaluated by using k-nearest neighbor classifier on land cover classification. Since collecting labels for geophysical image is very expensive, Gomez *et al.*'s work [27] and Kim *et al.*'s work [36] combine kernel machines and manifold learning method through regularization to exploit both labeled data and unlabeled data. Motivated in part by its success in geophysical image classification [27, 68] of specific interest in our work is the Laplacian Eigenmaps (LE) method for constructing low dimensional manifolds which preserve the local structure observed in the high dimensional space. Crucial to this process is the use of a well-designed weight factor which, for example in Yang *et al.*'s work [78], was used to build a manifold reflecting similarity in both the spectral and spatial components of the data for geophysical image classification.

While the majority of the work using manifold methods for geophysical machine learning has focused on the unsupervised case, when labeled data are available (as is the case here), one important supervised extension of LE was introduced in Perry *et al.*'s work [55], which combines the objective functions of LE and classifier training by regularization to incorporate the label in the manifold learning. We also note two other supervised LE methods proposed in Raducanu *et al.*'s work [56] and Wu *et al.*'s work [76] for face classification, where the weight factor is constructed to reflect similarity in examples from the

same class. In all of these cases, the labels were discrete-valued reflecting the fact that these were classification problems. For the regression problem of interest to us in this work, no straightforward adaptation of these methods exists. Our approach has been to include the label information (that is, knowledge of f_p, M_p, M_g for the training data) directly into the LE weight factors which is proposed in Section 4.1.

As explained more fully in [9], the Laplacian Eigenmaps procedure seeks a collection of length m manifold coordinate vectors $\mathbf{r}_i$ which minimize the objective function,

$$\sum_{i,j=1}^{N} \|\mathbf{r}_i - \mathbf{r}_j\|_2^2 \, \omega_{ij} = \mathrm{tr}(\mathbf{R}\mathbf{L}\mathbf{R}^T) \tag{2.15}$$

where $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N]$, $\mathbf{L} = \mathbf{D} - \mathbf{\Omega}$ is called the *Laplacian matrix*, $\mathbf{\Omega}$ is comprised of the ω_{ij}, here we define $\omega_{ij} = \exp(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma_1}) \exp(\frac{\|\mathbf{t}_i - \mathbf{t}_j\|_2^2}{\sigma_2})$ to measure the similarity between data. The matrix $\mathbf{D}$ is a diagonal matrix whose entries are $d_{ii} = \sum_{j=1}^{N} \omega_{ij}$. Formally, the optimization problem of LE is,

$$\begin{aligned} \min_{\mathbf{R} \in \Re^{m \times N}} \quad & \mathrm{tr}(\mathbf{R}\mathbf{L}\mathbf{R}^T) \\ \text{s.t.} \quad & \mathbf{R}\mathbf{D}\mathbf{R}^T = \mathbf{I}. \end{aligned} \tag{2.16}$$

where the constraint $\mathbf{R}\mathbf{D}\mathbf{R}^T = \mathbf{I}$ is added to eliminate the trivial solution $\mathbf{R} = \mathbf{0}$. Standard Lagrange multiplier method can be used to solve this problem. The Karush-Kuhn-Tucker optimality condition requires that the optimal $\mathbf{r}_i$ are obtained via the generalized eigen-decomposition problem

$$\mathbf{L}\mathbf{v}_k = \lambda_k \mathbf{D}\mathbf{v}_k \quad \text{where } \lambda_k, \ k = 1, 2, \cdots, m+1 \tag{2.17}$$

Specifically, we pick $m \ll d$ the eigenvectors corresponding to second smallest through $(m+1)^{th}$ eigenvalues with $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{m+1}$ because $\mathbf{v}_1$ is $\mathbf{1}$ vector which is useless for regression and the minimum value of objective function is $\sum_{k=1}^{m+1} \lambda_k$. The manifold coordinates of i^{th} datum are the i^{th} coefficients of all the eigenvectors, i.e., $\mathbf{r}_i = (v_{2,i}, v_{3,i}, \cdots, v_{m+1,i})^T$.

2.5 Out-of-Sample Extension

The manifold dimension reduction methods reviewed in the last section directly give the manifold coordinates of training data set, but the estimation of test data requires the embedding of test data in the same manifold as training data set. In this section, we review the out-of-sample extension algorithm for the manifold dimension reduction. Bengio *et al.*'s work [10] proposed to extent the manifold embedding to new data using Nyström formula because all the manifold dimension reduction methods can be converted to the eigen-decomposition problem of similarity matrix $\mathbf{M} \in \Re^{N \times N}$ of training data.

$$\mathbf{M}\mathbf{v}_k = \lambda_k \mathbf{v}_k \quad \text{where } \lambda_k,\ k = 1, 2, \cdots, N \tag{2.18}$$

Each coefficient M_{ij} of matrix $\mathbf{M}$ is the kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ used to measure the similarity between training data. For our problem $k(\mathbf{x}_i, \mathbf{x}_j) = \omega_{ij}$. The manifold coordinates of ith training datum are composed of all the ith elements of eigenvectors $\{\mathbf{v}_k\}$ in (2.18). The computational cost of eigen-decomposition of (2.18) is $\mathcal{O}(N^3)$, it is an issue for huge data set where N is very large. The solution of this problem is the subsampling of training data set to construct a smaller similarity matrix $\mathbf{M}' \in \Re^{n \times n}$ where $n < N$, thus the computational cost will be reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(n^3)$. When we want to compute the manifold coordinates

for the rest set of training data, the Nyström formula can be applied [63].

Shawe-Taylor *el at.*'s work [75] gives the k-th element of manifold coordinates $\mathbf{r}$ for the datum $\mathbf{x}$ in the rest of training data set as,

$$r_k = \frac{\sqrt{n}}{\lambda_k} \sum_{i=1}^{n} v_{ki} k(\mathbf{x}, \mathbf{x}_i) \tag{2.19}$$

where λ_k is the k-th eigenvalue of the small similarity matrix $\mathbf{M}'$ for subsampling training data set, v_{ki} is the i-th element of k-th corresponding eigenvector, $k(\cdot, \cdot)$ is the kernel used to calculate the similarity between data and $\mathbf{x}_i$ is the ith training datum. The drawback of this algorithm is that the manifold coordinate of new datum depends on the kernel used to calculate the weight factors in similarity matrix, thus the kernel can not include label information if we apply Nyström formula to embed the test data in the manifold.

Since the manifold dimension reduction methods give the manifold coordinates of training data, with raw feature and the manifold representation pair $\{(\mathbf{x}_i, \mathbf{r}_i),\ i = 1, \cdots, N\}$, another idea of learning the embedding function from raw feature space to manifold space using Spectral Regression is discussed in Section 4.2.

2.6 Data Sets

In order to evaluate the performance of our classification and regression approaches, we utilize the data generated from a field-scale simulation library. This library was designed to provide a variety of source zone architectures resulting from various spill and hydraulic conductivity scenarios, and then the library was used to identify source zone metrics controlling plume evolution. Two different hydraulic conductivity field models were used to generate

the simulation data, one is Sequential Gaussian Simulation (SGS) geo-statistical methods [42] conditioned to the Bachman, Michigan site [2], the other is Transition Probability based Markov Chain (TP/MC) model [48]. In the following two sections, we introduce these two types of data sets.

2.6.1 SGS Data

The simulation methodology for DNAPL infiltration and subsequent mass dissolution in the saturated zone has been reported before by Christ *et al.*'s works [20, 19, 22]. Briefly, DNAPL infiltration, entrapment and dissolution were simulated for three-dimensional (3D) non-uniform hydraulic conductivity fields based upon the Oscoda, Michigan site [42]. Three simulated data sets were considered for the evaluation of the methods in this work. DNAPL infiltration and entrapment were simulated with UTCHEM 9.0 using a baseline hydraulic conductivity field representative of a relatively homogeneous glacial out-wash deposit.

In data set-1, we combined three different spill scenarios (DNAPL infiltration, entrapment and dissolution). Scenario-1 of data set-1 in Table 2.1 is the *baseline* and was reported before in Christ *et al.*'s work [22]. Here an ensemble of 16 equally probable realizations of 3D hydraulic conductivity field was obtained from Lemke *et al.*'s work [42]. The baseline scenario consisted of a release of 128 liters of PCE for a period of 400 days located in a 4×5 grid area centered in the top layer of the domain. Following infiltration and entrapment, MT3DMS [80] was used to simulate dissolution under natural gradient conditions and the source zones with their correspondent down-gradient transect (end section of the domain) concentration data were recorded at every 20 time steps. Thus each dissolution simulation produced nearly 20 time-dependent plume response signals for machine learning approaches.

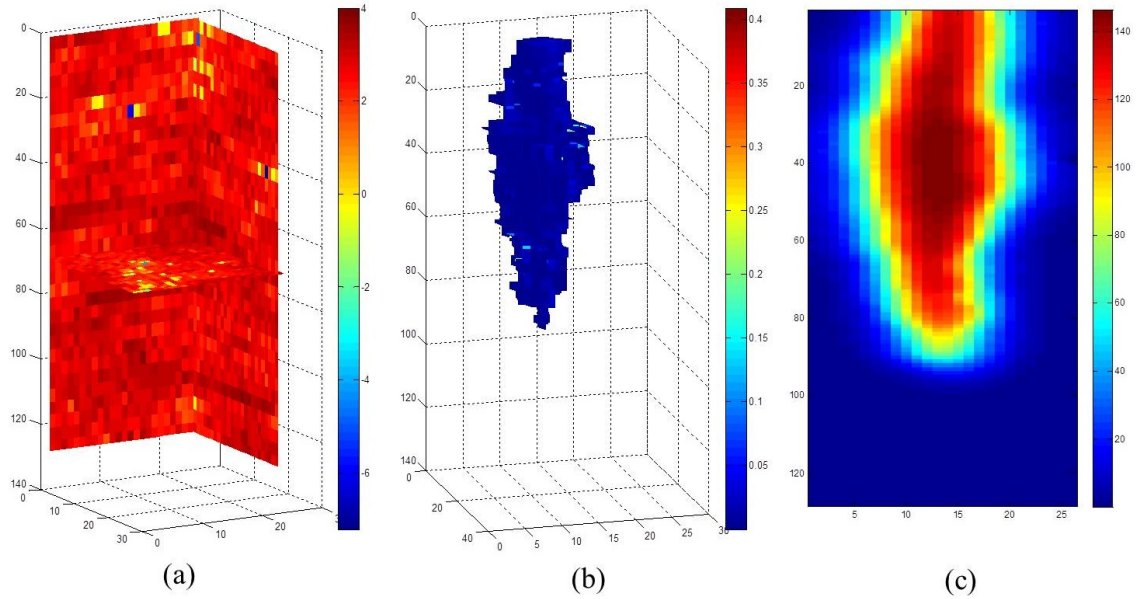

Figure 2.1: The results of SGS data, subfigure (a) shows the hydraulic conductivity in three dimension $26 \times 26 \times 128$, in which the values are presented in log scale. Subfigure (b) shows the saturation of PCE, the regions with the saturation lower than 0.15 are ganglia and the regions with the saturation higher than 0.15 represent the pools. Subfigure (c) is the concentration image, the colorbar shows the range of concentration value (mg/L), the shape of concentration is obviously closely related to the shape of saturation in source zone.

One set of samples of hydraulic conductivity field, the saturation in the source zone and the downstream concentration profile are shown in Figure 2.1. In Figure 2.1(a), the hydraulic conductivity is very high in source zone, the saturation of PCE at most parts is low which means the ganglia region dominates the source zone, therefore the down-gradient concentration image has very high values.

Scenario-2 and 3 of data set-1 are variations of the baseline scenario. Scenario-2 consisted of a catastrophic release, with the same release volume and location as the baseline scenario but a release period of 4 days. Scenario-3 of data set-1 consisted of the same volume and release period as the baseline but a different release location. In this scenario the release was located in the top layer of the 3D domain in 2 block areas: one block of 3×4 grid area

Matrix Properties	Data set 1			Data set 2		Data set 3	
Variogram Parameters	H	V		H	V	H	V
Nugget	0.333	0.333		0.333	0.333	0.333	0.333
Range (m)	7	1.07		7	1.07	7	1.07
Integral Scale (m)	2.33	0.36		4.66	0.72	4.66	0.72
σ^2 $(ln(k))$	0.29			1		1.5	
Mean Hydraulic Conductivity, (m/day)	16.8			16.8		16.8	
Anisotropy Ratio kv/kh	0.5			0.5		0.5	
Spill conditions	Scenario1	Scenario2	Scenario3				
Spill Volume (L)	128	128	128	128		128	
Spill Duration (d)	400	4	400	400		400	

Table 2.1: Conditions of the 3 different data sets generated for the machine learning algorithm implementation.

located north from the center and the other of 4×2 located south from the center.

The spill scenario used in scenario-1 of data set-1 was used for data set-2 and 3. The difference in these sets rested on the statistical properties used for obtaining the hydraulic conductivity realizations. As shown in Table 2.1, modifications to the statistics included longer correlation length values for data set-2 and 3, and higher log scale transformed hydraulic conductivity variance of 1.0 and 1.5 for data set-2 and 3 respectively. In order to test the performance of regression function under a wide range of conditions, we also combine these three data sets as data set-4. Taken together, these simulations provide a large library comprised of scenarios with different spill release history, spill configuration and heterogeneity of hydraulic properties.

2.6.2 Markov Chain Model

In addition to the Sequential Gaussian Simulation data sets, simulations with hydraulic conductivity fields based upon a highly heterogeneous glaciofluvial deposit [49] were performed to investigate the influence of the capillary pressure saturation (Pc-sat) parameters, residual organic saturation, spill rate and hypothetical field structure. Two dimensional hydraulic conductivity realizations were generated using a transition probability based Markov chain (TP/MC) approach. The glaciofluvial deposit hydraulic conductivity realizations were characterized by four dominant lithofacies with a high degree of continuity in the horizontal versus the vertical direction [49].

In the simulated scenarios, the soil matrix properties were generated according to the transition probability-based Markov Chain hydraulic conductivity distribution, representative of highly heterogeneous glaciofluvial deposits. The hydraulic conductivity region was modeled following the aquifer located 500 meters west of town Herten, southwest of German [32, 7, 37]. There are four dominant lithofacies with increasing hydraulic conductivity that are shown in Figure 2.2(a) from 1 to 4. The properties of these lithofacies are listed in Table 2.3 [49]. The lithofacies pattern of hydraulic conductivity field in each direction is determined by transition probability based Markov Chain model, the type of lithofacies at next location depends only on the type of lithofacies at current location. In the x direction, for an example, the transition probability p_{ij} from lithofacies component i at current location x to the component j at next location $x + \delta x$ is defined as,

$$p_{ij} = \Pr(\text{component } j \text{ at } x + \delta x | \text{component } i \text{ at } x) \tag{2.20}$$

the transition matrix for each direction is reported from Maji *et al.*'s work [48] which is in Table 2.4.

The hydraulic conductivity fields were generated in the following way. First, an unconditional categorical simulation was generated based on the TP/MC model obtained from laboratory-scale region. Second, the categorical data were sampled from a set of 22 randomly located points from the unconditional simulation. Finally, a set of hydraulic conductivity fields was conditionally generated using the TP/MC model and the 22 randomly located data points.

In these scenarios, MVALOR-2D was used to simulate the infiltration and entrapment of TCE and modified MT3D was applied for dissolution in two dimensional aquifer cells. The parameter settings of TCE infiltration is summarized in Table 2.2. Pc-sat properties were obtained from Schroth *et al.*'s work [62]. To generate an ensemble of hydraulic conductivity field realizations for these numerical simulations, the hydraulic conductivity field (aquifer cell) was discretized into 41×91 grid nodes with grid dimensions of 0.025×0.005 m. This simulation was then sampled using a set of 25 randomly located points. These sparse concentration signals are used for application and modification of the manifold regression approaches in Chapter 5. A sample set of hydraulic conductivity, saturation and concentration is shown in Figure 2.2. Figure 2.2(a) represents the hydraulic conductivity in which four components are indicated by number from 1 to 4 with increasing hydraulic conductivity. The DNAPL in Figure 2.2(b) was spilled on the top of 2D model and resided with high saturation above the low hydraulic conductivity layer. The flow dissolved the DNAPL and transported from left to right, the concentration was observed on the right side of 2D model in Figure 2.2(c).

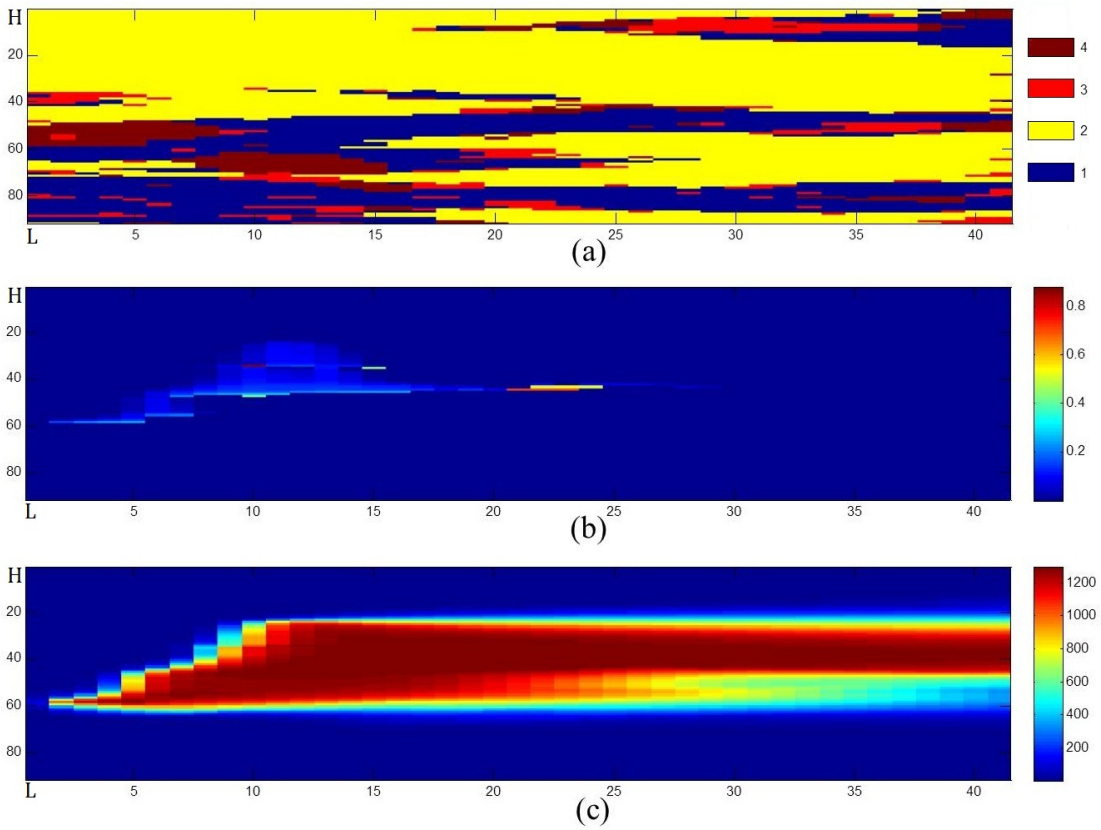

Figure 2.2: The result of TP/MC data, subfigure (a) is the hydraulic conductivity field, in which four components have different hydraulic conductivity, they are represented from 1 to 4 with the increasing hydraulic conductivity. Subfigure (b) shows the saturation of TCE which is spilled on the left side. The flow transport from left to right, subfigure (c) is the concentration and the colorbar shows the range of concentration value (mg/L).

Fluid Properties	Water	TCE
Density (g/cm^3)	0.999	1.470
Dynamic viscosity (cP)	1.14	0.59
Compressibility (Pa^{-1})	4.4×10^{-10}	0.0
Initial saturation	1.0	0.0
Spill conditions		
Spill Volume (ml)	39.6	
Spill Duration (hr)	1.32	
Redistribution time (hr)	22.66	
Release Rate (ml/hr)	30	
Spill Location (cm)	(x-25,z-33)	

Table 2.2: The parameter settings for simulation of infiltration of TCE.

No.	Lithofacies	Description	Volume Percentage	Hydraulic Conductivity (m/s)	Porosity
1	Gs-x	well-sorted gravel	29%	4.30×10^{-5}	0.2
2	Gcm	poorly-sorted gravel	57%	2.30×10^{-4}	0.23
3	S-x	pure, well-sorted sand	6%	1.00×10^{-3}	0.24
4	bGcm	cobble and boulder rich gravel	6%	8.00×10^{-2}	0.26

Table 2.3: The properties of lithofacies components in Herten site, southwest German.

	Gs-x	Gcm	S-x	bGcm
	x direction			
Gs-x	3.6	0.17	0.53	0.30
Gcm	0.15	14.6	0.51	0.33
S-x	0.41	0.3	0.78	0.29
bGcm	0.65	0.04	0.31	1.12
	y direction			
Gs-x	7.1	0.002	0.51	0.48
Gcm	0.34	9.8	0.51	0.15
S-x	0.48	0.33	1.5	0.19
bGcm	0.31	0.39	0.3	2.2
	z direction			
Gs-x	0.3	0.37	0.43	0.20
Gcm	0.53	0.90	0.30	0.17
S-x	0.71	0.29	0.10	0.001
bGcm	0.66	0.33	0.01	0.2

Table 2.4: The transition matrix of different lithofacies components, the diagonal entries of each direction are the average thickness of lithofacies components (m), the off-diagonal entries are the transition probability.

Chapter 3

Classification

In this chapter, first we propose a set of new features motivated by the hydrological model introduced in Section 2.1. The morphological image processing method is applied to extract the features from concentration image which is predictive for the estimation of metrics (i.e., f_p, M_p and M_g) in source zone. Therefore this set of features is used for both classification and regression framework. Second, we propose our classification approach to estimate a metric interval for each concentration datum represented by feature vector $\mathbf{x}$. Taking the classifying the pool fraction as an example,

$$\hat{f}_p = \mathcal{C}(\mathbf{x}) = \begin{cases} \mathcal{C}_1 & \hat{f}_p \in [0, B_1) \\ \mathcal{C}_2 & \hat{f}_p \in [B_1, B_2] \\ \mathcal{C}_3 & \hat{f}_p \in (B_2, 1] \end{cases} \tag{3.1}$$

where $\mathcal{C}$ is the classifier which assigns one label to an interval of pool fraction (e.g. $\mathcal{C}_1$ means the pool fraction is between 0 and B_1). The B_1 and B_2 are the boundaries between classes. The classifier training process and the testing procedure are shown in Figure 3.1.

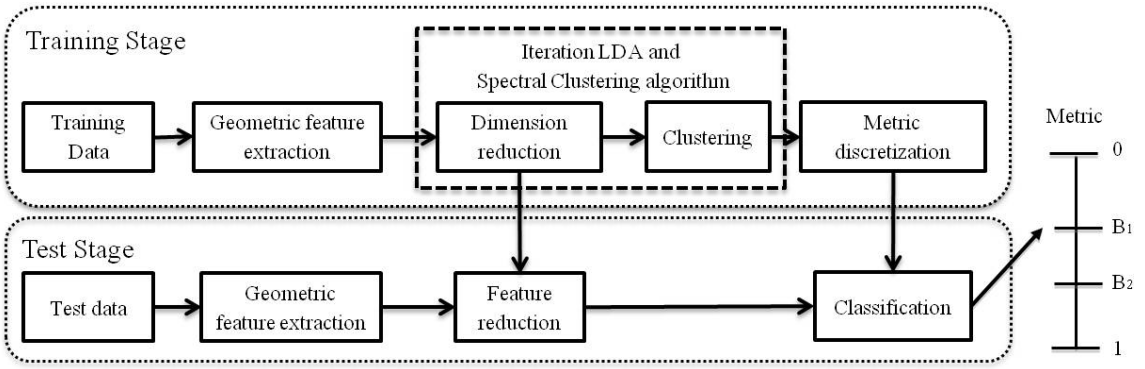

Figure 3.1: The framework of our classification-based machine learning approach.

After feature extraction, as discussed in Section 1.2.2, we need to reduce the dimension of these training data and cluster these data in this reduced feature space. We introduce the classic Principle Component Analysis and K-means method (PCA and K-means algorithm) in Section 3.2 for the dimension reduction and clustering steps respectively in Figure 3.1. These algorithms are used for comparison against our proposed iterative Linear Discriminant Analysis and Spectral Clustering algorithm (LDA-SC algorithm). Since the metrics are continuously valued, after discretization we divide the metric into several non-overlapping bins, each of which represents a class. The k-nearest-neighbor method is employed to classify the test datum. The metric discretization and k-nearest-neighbor classifier are used for both PCA and K-means algorithm and LDA-SC algorithm. In the experiment section, we compare the performance of our LDA-SC algorithm to the PCA and K-means algorithm, which demonstrates the superior classification ability of LDA-SC approach.

3.1 Geometric Feature Extraction

As shown in Figure 1.1, we consider a scenario in which we are provided observations of contaminant concentrations within a transect located away from the source zone and

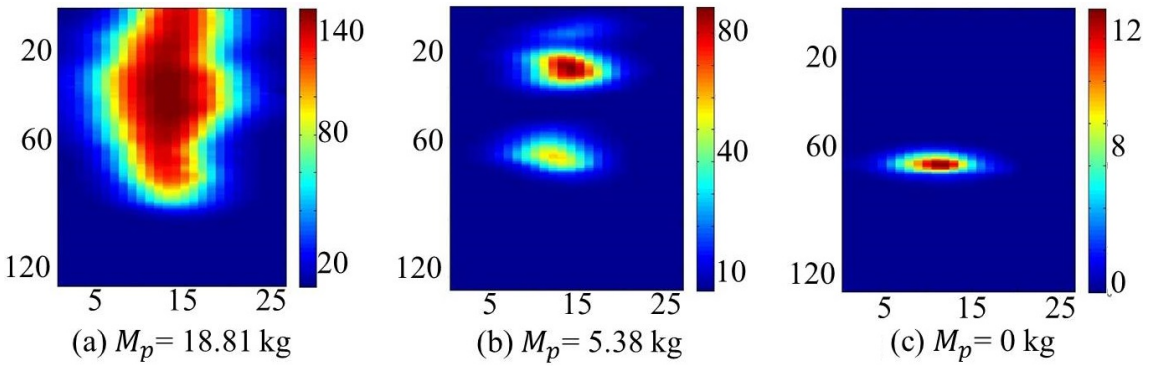

Figure 3.2: The observation of concentration image according to their mass in pools value.

oriented orthogonally to the nominal direction of groundwater flow. The feature vector we develop is motivated by our intuition concerning how the morphology of the observed concentration data is related to that of the unknown DNAPL saturation in the source zone. As an example, we consider the concentration data and associated mass in pools (i.e., similar intuition can also be established for the other two metrics f_p, M_g) shown in Figure 3.2. Roughly speaking we observe that as the mass in pools decreases, the geometry of the concentration data changes accordingly. Specifically the number of "blobs" in the images increases and their sizes decrease. Motivated by this observation, here we seek the features that capture the size and number of blobs in the concentration data believing that they are related to the metrics in a way that can be learned given sufficient examples.

To motivate the mathematical definitions for the feature vector we develop, in Figure 3.3 we display samples of the concentration data as height maps along with the corresponding geometric feature vectors for the same data displayed as images in Figure 3.2.

Mathematically, the key issues here are quantifying the notion of a "blob" and determining what characteristics of these blobs are useful. The first issue is addressed by a simple thresholding operation in which we specify the blobs at some level τ to be those pixels in

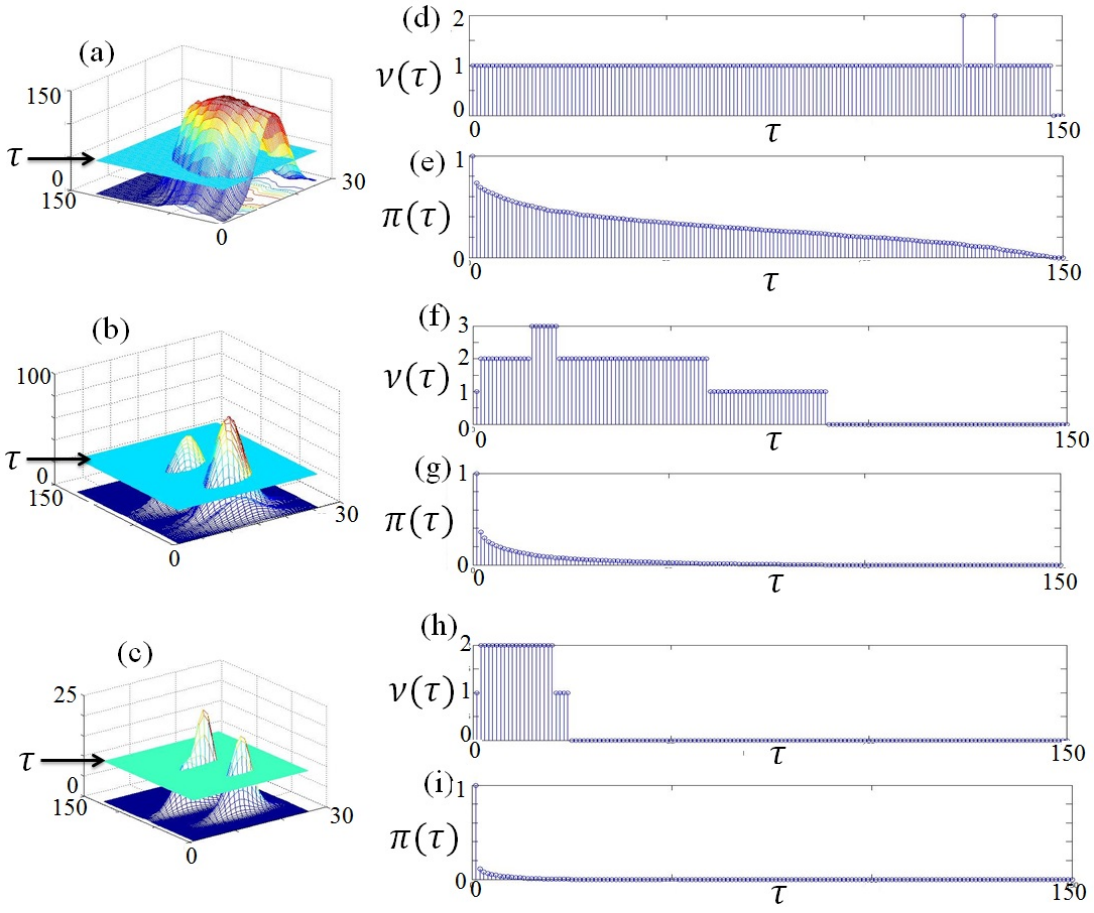

Figure 3.3: The geometric feature vectors of concentration data in Figure 3.2.

the image whose concentration values exceed τ :

$$b(x,y;\tau) = \begin{cases} 1 & \text{if } \mathbf{c}(x,y) > \tau \\ 0 & \text{else} \end{cases} \tag{3.2}$$

where (x, y) is the coordinate of concentration data. From $b(x, y; \tau)$ we have found it useful to compute two quantities: the percentage of the area in $\mathbf{c}(x, y)$ for which $b(x, y; \tau) = 1$ denoted as $\pi(\tau)$, and the number of connected components at that level, $\nu(\tau)$. The percentage of area calculation is,

$$\pi(\tau) = \frac{\sum_{x,y} b(x,y;\tau)}{\sum_{x,y} b(x,y;0)} \tag{3.3}$$

where the denominator is nothing more than the number of pixels in the concentration image that are nonzero. Referring to the data in Figure 3.3 again, for sub-figure(a) with $\tau = 20$, the number of connected components is one. For (b) at $\tau = 10$ there are two connected components while in (c) with $\tau = 5$ there are only two small size connected components.

The geometric feature vector we create, denoted as $\mathbf{x}$, is comprised of $\pi(\tau)$ and $\nu(\tau)$ for $\tau = 0, 1, 2, \cdots, \tau_{max}$ where τ_{max} is the largest value of concentration in the training data set. We define $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \Re^{d \times N}$ as the geometric feature matrix constructed using all data in our training set. In Figure 3.3(d) - (i), we plot $\pi(\tau)$ and $\nu(\tau)$ corresponding to concentration data in Figure 3.2(a) - (c). We see that the behavior of these quantities as a function of threshold is quite distinct depending on the M_p value. The number of connected component is almost always one in Figure 3.3(d) for the high mass in pools case,

indicating that there is only one large diffuse area in concentration image[1]. In Figure 3.3(f), the pattern is more variable reflecting that the structures in the concentration image are themselves more complex. Finally in Figure 3.3(h) the number of connected components drops to zero quite quickly reflecting the presence of only a single blob in the concentration data due to the low mass in pools (i.e., a lack of diffuse ganglia in the saturation profile). Similarly, the decays of $\pi(\tau)$ as a function of τ illustrated in Figure 3.3(e), (g) and (i) show a strong dependence on the underlying mass in pools.

3.2 PCA and K-means

In this section, we introduce the widely applied unsupervised dimension reduction method PCA and the K-means clustering method (PCA and K-means algorithm). These algorithms will be applied for comparison against our iterative LDA-SC algorithm which is proposed in Section 3.3. As discussed in Section 1.3.1, we need first to reduce the dimension of geometric feature vectors proposed in Section 3.1, and then cluster the reduced feature vectors. Since the metric value is continuous, we do not have class labels for training data set, we apply unsupervised dimension reduction method, Principle Component Analysis (PCA), to reduce the dimensionality of training features. PCA can find the most representative linear low dimensional subspace to embed the original high dimensional data without the label

[1] Although there is some variability in feature vector (e.g. the number of connected component is two for a couple of larger values of the threshold) such inconsistencies have little impact in the ultimate utility of these features.

information. It minimizes the distortion function below,

$$\begin{aligned} \min_{r_{ik},\mathbf{e}_k} \quad & J = \sum_{i=1}^{N} \| \sum_{k=1}^{m} r_{ik}\mathbf{e}_k + \bar{\mathbf{x}} - \mathbf{x}_i \|_2^2 \\ \text{s.t.} \quad & \|\mathbf{e}_k\|_2^2 = 1, \quad k = 1, 2, \cdots, m. \end{aligned} \tag{3.4}$$

where $\mathbf{r}_i = [r_{i1}, r_{i2}, \cdots, r_{im}]^T$ is the reduced feature vector of $\mathbf{x}_i$ and $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$. The solution to minimize the above function is to determine first the orthogonal directions $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_m\}$ using the method discussed below and the set $r_{ik} = \mathbf{e}_k^T(\mathbf{x}_i - \bar{\mathbf{x}})$.

In order to solve the optimization problem (3.4), using linear algebra the orthogonal directions are the eigenvectors of the scatter matrix $\mathbf{M}$ of training data [35], which is

$$\mathbf{M}\mathbf{e}_k = \lambda_k \mathbf{e}_k, \quad k = 1, 2, \cdots, m, \quad \mathbf{M} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T. \tag{3.5}$$

When the distortion function J is minimized, $J = \sum_{k=m+1}^{d} \lambda_k$ and d is the dimension of geometric feature vector, we select $m = 6$ resulting in $\frac{\sum_{k=7}^{d} \lambda_k}{\sum_{k=1}^{d} \lambda_k} = 0.1$. The eigenvalues of covariance matrix $\mathbf{M}$ are the variances of data set in eigenvector directions. We select $m = 6$ such that the residual variance is only 10% of total variance of data set. For the test data, the reduced feature vector $\tilde{\mathbf{r}} = [\tilde{r}_1, \tilde{r}_2, \cdots, \tilde{r}_m]$ is calculated as $\tilde{r}_k = \mathbf{e}_k^T(\tilde{\mathbf{x}} - \bar{\mathbf{x}})$ where $k = 1, 2, \cdots, m$.

One of the widely applied clustering methods is the K-means technique, which is an iterative clustering algorithm and tries to minimize the criterion function below,

$$\min_{p_{ik},\bar{\mathbf{r}}_k} J = \sum_{i=1}^{N}\sum_{k=1}^{K} p_{ik} \|\mathbf{r}_i - \bar{\mathbf{r}}_k\|_2^2 \tag{3.6}$$

where $\bar{\mathbf{r}}_k$ represents the center of cluster $\mathcal{C}_k$, p_{ik} is the assignment indicator variable $p_{ik} = \{0, 1\}$, $p_{ik} = 1$ indicates that the $\mathbf{r}_i$ belongs to cluster $\mathcal{C}_k$.

The goal of K-means is to find optimal $\{p_{ik}\}$ values and calculate the center of each cluster $\{\bar{\mathbf{r}}_k\}$ according to the assignment indicators. The algorithm achieves this goal by iterating the following two successive steps. After initialing K cluster centers, in the first step we fix the centers and use Euclidian distance to calculate $\|\mathbf{r}_i - \bar{\mathbf{r}}_k\|_2^2$ for each data, then minimize J with respect to $\{p_{ik}\}$. The solution of $\{p_{ik}\}$ is below.

$$p_{ik} = \begin{cases} 1 & k = \text{argmin}_k \|\mathbf{r}_i - \bar{\mathbf{r}}_k\|_2^2, \quad k = 1, 2, \cdots, K \\ 0 & \end{cases} \tag{3.7}$$

In the second step, we minimize J with respect to $\{\bar{\mathbf{r}}_k\}$, keeping $\{p_{ik}\}$ fixed. The solution of $\bar{\mathbf{r}}_k$ is

$$\bar{\mathbf{r}}_k = \frac{1}{n_k} \sum_{\mathbf{r}_i \in \mathcal{C}_k} \mathbf{r}_i \tag{3.8}$$

where n_k is the number of data in cluster $\mathcal{C}_k$. We do these two successive steps iteratively until $\{\bar{\mathbf{r}}_k\}$ does not change. There is still one problem that we need to determine the number of clusters K beforehand, in the experiment section we set $K = 3$.

3.3 Linear Discriminant Analysis and Spectral Clustering

PCA method in the last section reduces the dimension of training features without considering the label information, but the objective of dimension reduction is to find a low dimensional space that the clusters can be separated so far away that the test datum can be classified easily into one of clusters. Therefore Linear Discriminant Analysis (LDA)

[17] is a more appropriate choice for dimension reduction because it is a well-established method for finding a linear transformation $\mathbf{T}$ that, in a precise mathematical sense, projects high dimensional feature vectors such as the geometric feature vectors $\mathbf{x}$, onto a reduced dimensional space in a manner that maximally separates classes, (e.g. the different bins of pool fraction, etc), that is the reduced feature $\mathbf{r} = \mathbf{T}\mathbf{x}$, where $\mathbf{r} \in \Re^m, \mathbf{T} \in \Re^{m \times d}$ and the geometric feature $\mathbf{x} \in \Re^d$. LDA needs the class labels which are given by clustering method. While K-means clustering method in the last section finds the clusters by minimizing the objective function (3.6), it can not guarantee the number of data in each cluster is "balanced". In our application, it is desired that the data are evenly distributed in each class because we use k-nearest-neighbor classifier which determines the label of test datum as the majority of class labels in its neighborhood, thus the classifier will not be bias to any class with balanced classes. The Spectral Clustering (SC) method can achieve this goal because in [71] SC method is proved to be equivalent to RatioCut problem, its objective function minimizes the similarity between classes and balances the distribution of data in each class. Thus in this section we propose an iterative algorithm based on a combination of LDA and SC to determine feature-based clusters that are associated with metric bins.

Assume training feature set $\mathbf{X}$ is partitioned into K clusters which is $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots \mathbf{X}_K]$. Two scatter matrices *within-class* $\mathbf{S}_w$ and *between-class* $\mathbf{S}_b$ are defined below [17],

$$\mathbf{S}_w = \frac{1}{N} \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} (\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T \tag{3.9}$$

where $\bar{\mathbf{x}}_k$ is the center of cluster $\mathcal{C}_k$, $\bar{\mathbf{x}}_k = \frac{1}{n_k} \sum_{\mathbf{x}_i \in \mathcal{C}_k} \mathbf{x}_i$, n_k is the number of data in cluster

$\mathcal{C}_k$.

$$\mathbf{S}_b = \frac{1}{N}\sum_{k=1}^{K} n_k(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^T \tag{3.10}$$

where $\bar{\mathbf{x}}$ is the center of whole training data set, $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$.

LDA intends to find the linear transform matrix $\mathbf{T}$ which can minimize the within-class distance and maximize the between-class distance simultaneously, thus the optimization problem is as following,

$$\mathbf{T}^* = \min_{\mathbf{T}} J = \frac{\mathrm{tr}(\mathbf{T}\mathbf{S}_w\mathbf{T}^T)}{\mathrm{tr}(\mathbf{T}\mathbf{S}_b\mathbf{T}^T)} \tag{3.11}$$

the solution of (3.11) is the eigenvectors of matrix $\mathbf{S}_w^{-1}\mathbf{S}_b$ for the eigenvalue $\lambda \neq 0$. In [25] it is shown that the number of non-zero eigenvalues is at most $K-1$. Thus $m = K-1 = 3-1 = 2$. The reduced feature space for training data is comprised of $\mathbf{r} = \mathbf{T}^*\mathbf{x}$ and the test data $\tilde{\mathbf{x}}$ can be projected in the same space by $\tilde{\mathbf{r}} = \mathbf{T}^*\tilde{\mathbf{x}}$.

The Spectral Clustering method clusters the training data in the reduced feature space. First a graph Laplacian matrix is construct [71],

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{3.12}$$

where $\mathbf{W}$ is the weight matrix, of which the entries are,

$$w_{ij} = \exp(-\frac{\|\mathbf{r}_i - \mathbf{r}_j\|_2^2}{\sigma}) \tag{3.13}$$

w_{ij} measures the similarity between data. If $w_{ij} \neq 0$, data $\mathbf{r}_i$ and $\mathbf{r}_j$ are connected in the graph, thus these data should belong to the same cluster. If $w_{ij} \approx 0$, data $\mathbf{r}_i$ and $\mathbf{r}_j$

are disconnected in the graph, so these data should belong to the different clusters. σ is chosen by cross-validation [11] which is discussed in Section 3.6. $\mathbf{D}$ is a diagonal matrix whose entries are $d_{ii} = \sum_{j=1}^{N} w_{ij}$. The spectral analysis of the Laplacian matrix $\mathbf{L}$ is the eigen-decomposition,

$$\mathbf{L}\mathbf{v}_k = \lambda_k \mathbf{v}_k, \quad \text{where } 0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_K \tag{3.14}$$

Assume there are three non-overlapping clusters which means the datum is only connected to the data within its cluster and there is no connection between the data in different clusters. The Laplacian matrix L has a block diagonal form as follow [71],

$$\begin{pmatrix} \mathbf{L}_1 & & \\ & \mathbf{L}_2 & \\ & & \mathbf{L}_3 \end{pmatrix} \tag{3.15}$$

where $\mathbf{L}_k,\ k = 1, 2, 3$ is the Laplacian matrix for each cluster.

By definition, the sum of each row in Laplacian matrix is 0, thus the number of 0 eigenvalues of $\mathbf{L}$ is equal to the number of non-overlapping clusters [54], and the eigenvector $\mathbf{v}_k \in \Re^N$ corresponding to cluster $\mathcal{C}_k$ is filled with "1" at the position of data belonging to $\mathcal{C}_k$ and filled with "0" at other positions [53]. Since the clusters in our problem are not strictly non-overlapping, we take the K eigenvectors as $\mathbf{V} \in \Re^{N \times K}$ corresponding to the smallest K eigenvalues and apply K-means method to each row of $\mathbf{V}$.

The problem here is that the LDA method assumes that labeled data are available; i.e. that the bins into which the metric will be divided are known as a priori. As this is not the

case for our problem, we have developed an iterative algorithm (LDA-SC algorithm) that both finds the reduced dimension feature space and cluster the feature data in this space. The training procedure of the approach is summarized in the Table 3.1.

Algorithm: Linear Discriminant Analysis and Spectral Clustering iterative algorithm	
Inputs:	$\{\mathbf{x}_i, t_i\}_{i=1}^{N}$, and K the number of bins into which we wish to divide metric. $\mathbf{x}_i$ is the geometric feature vector and t_i is the metric
Outputs:	- K collections of feature vectors - An optimal linear transformation for projecting geometric feature vectors into a reduced feature space.
Initialization: Perform the Spectral Clustering to obtain initial class labels.	
Loop:	- LDA finds the reduced dimension space and the transform matrix $\mathbf{T}$. - Spectral Clustering in the low dimension subspace, updates the class labels. Change the labels of data to the new labels determined in this loop. - If the labels do not change or the number of loop exceeds 20, stop loop.

Table 3.1: The iterative LDA-SC algorithm to find the reduced dimension feature space and cluster the feature vectors in this space.

3.4 Metric Discretization

After clustering in the reduced feature space, we need to divide continuous metric value into intervals. Take f_p as an example, corresponding to each of these feature clusters is a grouping of associated pool fractions. Ideally, if the range of corresponding f_p value to each cluster does not overlap, it is easy to determine the boundary of f_p value. But, while the K groups of reduced feature vectors are disjoint, in general, the corresponding K groups of f_p will overlap. This is illustrated in Figure 3.4(a) for the case of two clusters where the cluster-1 and cluster-2 are disjoint in the reduced dimension feature space, but the corresponding ranges of metric are overlapping. To obtain non-overlapping bins then, we must choose a boundary in the overlap region of metric space. If, for example, we take the

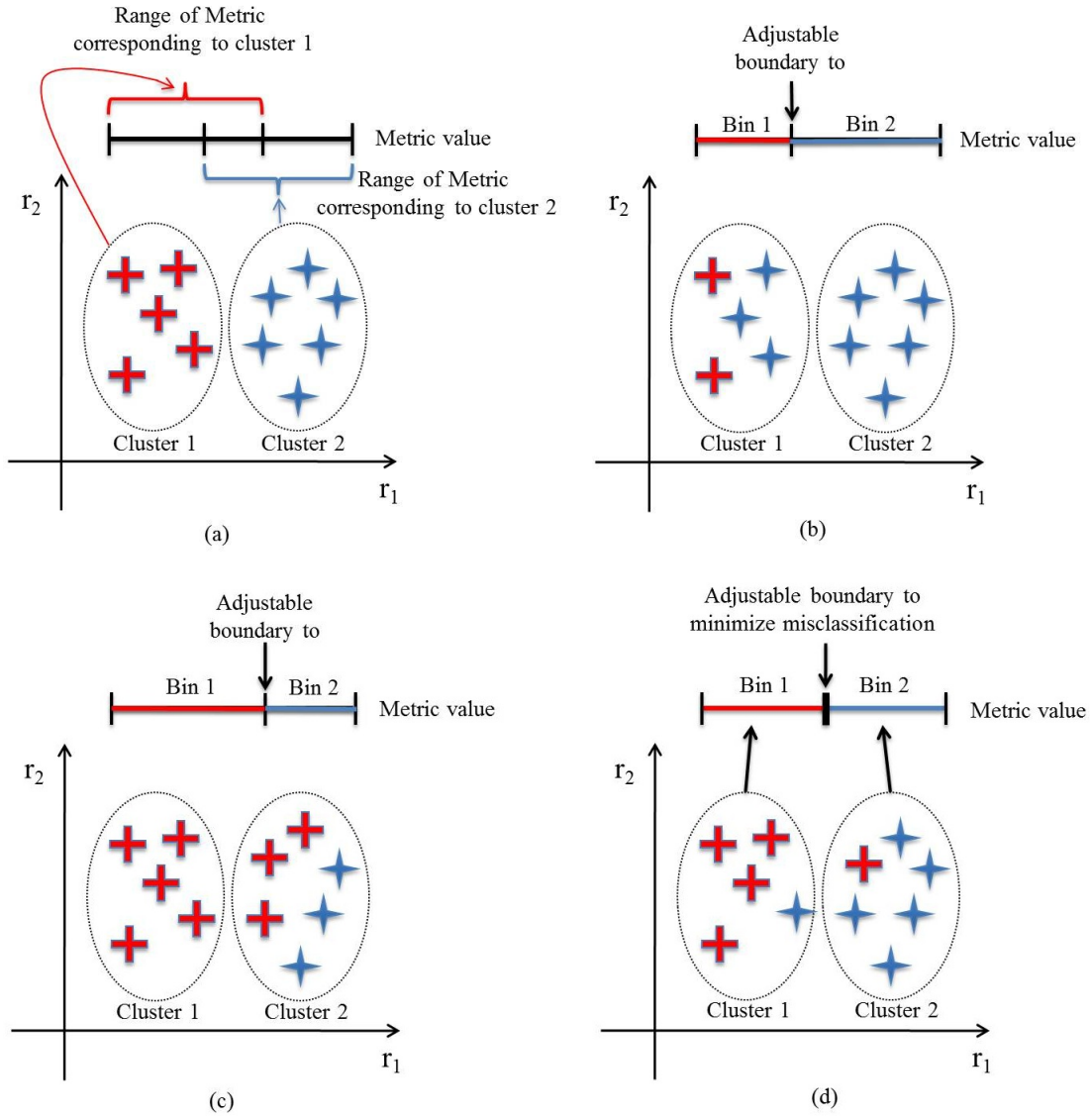

Figure 3.4: The illustration of boundary adjustment.

left extreme point of overlap region as the boundary, this would require that many of the red-cross training data would be "flipped" to blue stars. Hence this choice for the boundary would lead to a high misclassification rate among the training data in cluster-1 as shown in Figure 3.4(b). Likewise, if we take the right extreme point, the misclassification rate in cluster-2 will be high, as shown in Figure 3.4(c). Clearly, there is some choice between these two extremes for which the misclassification rate for the data in the training set is minimized. We search the range of overlap linearly to find the boundary to divide f_p value into intervals which can minimize the misclassification rate. For our example, these boundaries are displayed in Figure 3.4(d) with the black arrow. At the end of this process then, we have clusters of reduced feature vectors corresponding to non-overlapping bins in metric space.

3.5 k-nearest-neighbor Method

A k-nearest-neighbor (kNN) method is used as the basis for testing. This approach assumes the training data and test data live in the same space. The algorithm calculates the distance between test data and each training data to measure similarity, the smaller the distance is, the more similar the metrics of data are. The algorithm assigns the label of testing data as that of its nearest neighbors. As shown in Figure 3.5, the circle is the test data, the classifier takes $k = 5$ nearest neighbors to decide the label of test data. If the labels are not consistent, we use majority vote to determine the final label for test data, thus in Figure 3.5 the test data will be classified into class-1 and the metric interval corresponding to class-1 is determined by metric discretization. k-nearest-neighbor method is a decent way

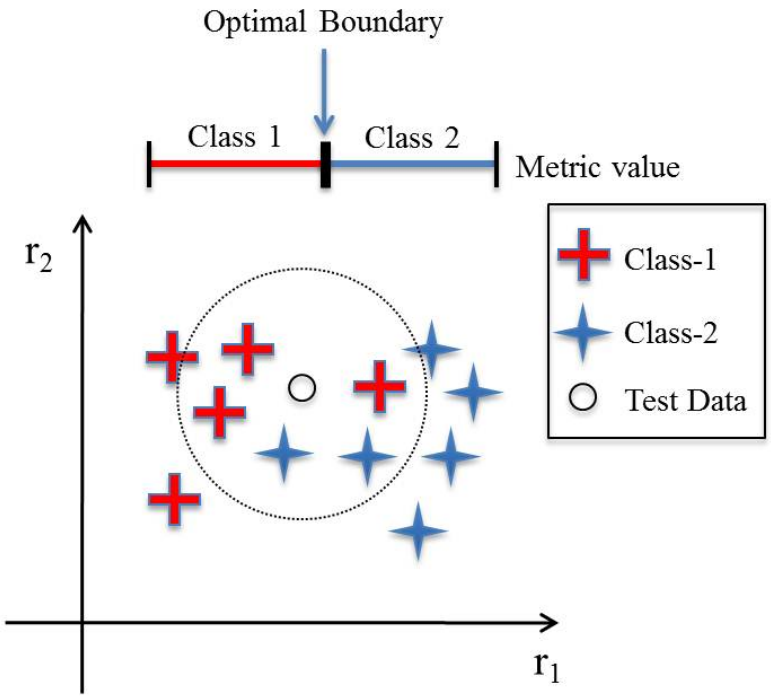

Figure 3.5: The illustration of k-nearest-neighbor method.

to build a classifier implicitly because we don not need to care about the shape of boundary between classes, it classifies the test data only depending on its neighborhood which gives an outstanding performance.

3.6 Experiments

In this section, we examine the performance of the LDA-SC algorithm using data set-1 which we introduce in Section 2.6.1, comparing to the result using PCA and K-means algorithm. There are three hyper-parameters we need to determine beforehand for the experiments. The first one is the number of classes K, if we choose $K = 1$, we classify all the test data into one class, the accuracy will be 100%, but the classification is meaningless; on the other hand, if we choose $K = N$, the number of classes is too many and the interval of metric

for each class is extremely narrow. Therefore the appropriate K needs to be determined according to the precision we want to achieve. In this section, we choose $K = 3$. The other two hyper-parameters are the σ in weight factor of Laplacian graph and the number of neighbors in k-nearest-neighbor method. The σ in (3.13) has an impact on the similarity measured between data, if σ is too small, almost all the weights are zero which means all the data are disconnected; if σ is too large, almost all the weights are one which means all the data are connected in Laplacian graph, thus the choice of σ will affect the result of clustering. For k-nearest-neighbor method, if $k = 1$ the test datum is classified based upon the label of its nearest neighbor, the classifier will not robust; whereas if $k = N$ all the test data are classified as the label of the majority in training data which makes the classifier useless. We use grid search to select σ and k manually, we run the experiments for each combination of σ and k, then select the σ and k when the accuracy of classification for test data is highest. From this process, the resulting values are $\sigma = 15$ and $k = 5$.

For each metric (f_p, M_p and M_g), we use 90% of data set for training and 10% of data set for test. The algorithm (PCA and K-means or LDA-SC) is applied to determine the boundary of metric, which is denoted as *one run* classification. The classification accuracy of class-k is calculated as,

$$\text{Accuracy} = \frac{\text{the number of test data that are classified correctly as class-}k}{\text{the number of test data in class-}k} \tag{3.16}$$

we also calculate the average accuracy as,

$$\text{Average Accuracy} = \frac{\text{the number of test data that are classified correctly}}{\text{the total number of test data in class}} \tag{3.17}$$

In order to estimate the classification accuracy of LDA-SC algorithm, we use the determined boundary in one run to label the training data and run 10-fold cross-validation using these boundaries [11] to estimate the true accuracy of each class. Since the label of each training data is fixed, we only need to perform LDA to find the reduced feature space and then use k-nearest-neighbor classifier to classify each test data. We take the mean of ten classification results as the estimation and ± 1.96 of standard deviation as the 95% confidence interval estimation which are shown as bold face number on the diagonal line in *confusion matrix* in Table 3.4, 3.7 and 3.10. The misclassification rates on the off-diagonal lines are calculated as,

$$\text{Misclassification Rate} = \frac{\text{the number of test data in class-k that are classified as class-}j}{\text{the number of test data in class-}k} \tag{3.18}$$

The one run classification results and confusion matrix of each metric are provided in Table 3.2 to 3.10. Since the PCA and K-means algorithm can not find the appropriate boundaries, we only show the confusion matrices for the LDA-SC algorithm.

Pool Fraction	Average Accuracy: 0.790		
	Class-1	Class-2	Class-3
Boundary	$0 < f_p < 0.49$	$0.49 < f_p < 0.58$	$0.58 < f_p < 1$
Accuracy	0.935	0.481	0.513
The number of training data	289	40	107

Table 3.2: The one run classification result of pool fraction for data set-1 using PCA and K-means algorithm.

In Table 3.2, the one run classification result of f_p using PCA and K-means algorithm is shown, more than half of data set is clustered as class-1 and the f_p interval corresponding to class-2 is very narrow (i.e., only 0.09), the accuracy of class-2 and class-3 is 0.481 and

Pool Fraction	Average Accuracy: 0.868		
	Class-1	Class-2	Class-3
Boundary	$0 < f_p < 0.36$	$0.36 < f_p < 0.56$	$0.56 < f_p < 1$
Accuracy	0.915	0.812	0.857
The number of training data	190	116	130

Table 3.3: The one run classification result of pool fraction for data set-1 using LDA-SC algorithm.

Pool Fraction		Estimated Class		
		Class-1	Class-2	Class-3
True Class	Class-1	$\mathbf{0.873 \pm 0.064}$	0.057 ± 0.045	0.070 ± 0.037
	Class-2	0.116 ± 0.044	$\mathbf{0.860 \pm 0.058}$	0.024 ± 0.036
	Class-3	0.134 ± 0.120	0.036 ± 0.034	$\mathbf{0.830 \pm 0.133}$

Table 3.4: The confusion matrix of pool fraction classification result for data set-1 using the boundaries found by LDA-SC algorithm.

0.513 respectively, the classifier randomly guesses the labels of test data in these two classes. Therefore, PCA and K-means algorithm does not work for the f_p classification. However, the performance of LDA-SC algorithm in Table 3.3 is much better, the average accuracy increase from 0.790 to 0.868, the number of data in each class are almost evenly distributed and the accuracy of each class is higher than 0.800. The confusion matrix of accuracy estimation for f_p is in Table 3.4. The estimated accuracy of each class is around 0.85.

Mass in Pools	Average Accuracy: 0.830		
	Class-1	Class-2	Class-3
Boundary (kg)	$0.59 < M_p < 6.55$	$6.55 < M_p < 28.7$	$28.7 < M_p < 44.6$
Accuracy	0.660	0.920	0.680
The number of training data	60	284	92

Table 3.5: The one run classification result of mass in pools for data set-1 using PCA and K-means algorithm.

The one run classification of M_p using PCA and K-means algorithm is in Table 3.5. The average accuracy reaches 0.830 because the accuracy of class-2 is very high which is 0.92, but the accuracy of class-1 and class-2 is around 0.650, barely better than the random guess.

Mass in Pools	Average Accuracy: 0.856		
	Class-1	Class-2	Class-3
Boundary (kg)	$0.59 < M_p < 9.26$	$9.26 < M_p < 26.7$	$26.7 < M_p < 44.6$
Accuracy	0.911	0.844	0.830
The number of training data	114	177	145

Table 3.6: The one run classification result of mass in pools for data set-1 using LDA-SC algorithm.

Mass in Pools		Estimated Class		
		Class-1	Class-2	Class-3
True Class	Class-1	$\mathbf{0.908 \pm 0.062}$	0.026 ± 0.026	0.066 ± 0.045
	Class-2	0.042 ± 0.044	$\mathbf{0.870 \pm 0.071}$	0.087 ± 0.053
	Class-3	0.079 ± 0.126	0.077 ± 0.099	$\mathbf{0.844 \pm 0.067}$

Table 3.7: The confusion matrix of mass in pools classification result for data set-1 using the boundaries found by LDA-SC algorithm.

The metric interval of class-2 is from 6.55 kg to 28.7 kg which almost two thirds of data set belong to, the kNN classifier is clearly biased to classify the test data into class-2. However, by using LDA-SC algorithm in Table 3.6, the number of data in each class is almost the same and all the accuracy is higher than 0.800 even though the average accuracy is only 0.02 higher than PCA algorithm. The confusion matrix in Table 3.7 shows the estimated accuracy is around 0.85.

Mass in Ganglia	Average Accuracy: 0.940		
	Class-1	Class-2	Class-3
Boundary (kg)	$0 < M_g < 14.1$	$14.1 < M_g < 30.5$	$30.5 < M_p < 201.4$
Accuracy	0.952	0.708	0.965
The number of training data	153	35	248

Table 3.8: The one run classification result of mass in ganglia for data set-1 using PCA and K-means algorithm.

The classification result of M_g is better than f_p and M_p because the shape of concentration is highly related to the ganglia region as shown in Figure 2.1. The average accuracy of PCA and K-means algorithm is 0.94 in Table 3.8 because the cluster-3 dominates the

Mass in Ganglia	Average Accuracy: 0.886		
	Class-1	Class-2	Class-3
Boundary (kg)	$0 < M_g < 13.4$	$13.4 < M_g < 69.1$	$69.1 < M_p < 201.4$
Accuracy	0.920	0.835	0.884
The number of training data	145	139	152

Table 3.9: The one run classification result of mass in ganglia for data set-1 using LDA-SC algorithm.

Mass in Ganglia		Estimated Class		
		Class-1	Class-2	Class-3
True Class	Class-1	$\mathbf{0.934 \pm 0.018}$	0.064 ± 0.021	0.002 ± 0.002
	Class-2	0.079 ± 0.019	$\mathbf{0.820 \pm 0.013}$	0.101 ± 0.002
	Class-3	0.000 ± 0.000	0.113 ± 0.017	$\mathbf{0.887 \pm 0.017}$

Table 3.10: The confusion matrix of mass in pools classification result for data set-1 using the boundaries found by LDA-SC algorithm.

training data set and the k-nearest-neighbor classifier is bias to classify the data into class-3 which is undesirable for classification. Although the average accuracy of LDA-SC algorithm decrease to 0.886, the accuracy of class-2 increases from 0.708 to 0.835 while the accuracy of class-1 and class-2 doesn't deteriorate much in Table 3.9. The confusion matrix in Table 3.10 shows the confidence intervals are less than 0.02 which means the classification results of M_g are very stable.

From the experiments, the performance of LDA-SC algorithm is much better than PCA and K-means algorithm. The reason is that LDA incorporates the label information into the dimension reduction, thus in the reduced feature space the between class distances are maximized and within class distances are minimized, this gives advantage to classification while PCA is an unsupervised dimension reduction method which has no benefit. Spectral Clustering method balances the distribution of data in each class, so the k-nearest-neighbor classifier is not bias to any class, this also gives a better performance than K-means method.

Chapter 4

Manifold Regression

Motivated by the ideas in Guo *et al.*'s work [30], here we present a regression based machine learning algorithm whose component steps are illustrated in Figure 4.1. As shown in Figure 4.1, the training and testing processes require a number of steps which we outline here. The inputs to the training phase are observations of down-gradient concentration data $\mathbf{c}_i(x, y)$ and an associated metrics vector $\mathbf{t}_i = [f_p, M_p, M_g]^T$ for the known source zone associated with these concentration signals with $i = 1, 2, \cdots, N$. The variables x and y represent the coordinates in the down-gradient transect where the concentration data are collected. Thus we may regard $\mathbf{c}_i(x, y)$ as an image where x and y index the coordinates of the pixels. It is also convenient to think of these data as a vector, $\mathbf{c}_i$, obtained by lexicographically ordering the pixels in the image. Thus N_c, the dimensionality of $\mathbf{c}_i$, is equal to the product of the number of rows and columns in $\mathbf{c}_i(x, y)$. As explained in Section 2.1 these training data are generated via numerical simulation where a number of DNAPL infiltrations are obtained followed by dissolution to generate the concentration data.

From the concentration data, the first step in training is geometric feature extraction

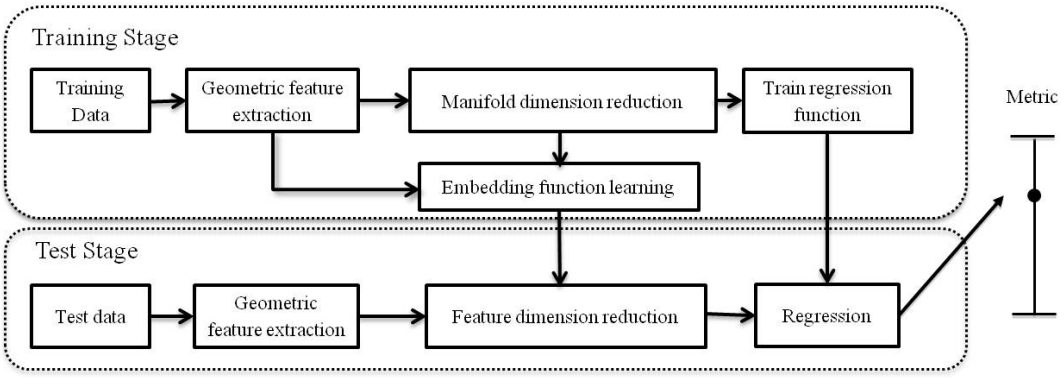

Figure 4.1: The framework of our regression-based machine learning approach.

using the same method proposed in Section 3.1. Then we employ manifold dimension reduction method to obtain the low dimensional manifold coordinate vectors for training regression functions under a Bayesian approach. For testing, we first learn the embedding function, which is a mapping from geometric feature space to manifold space, using training data. Then we can embed the test data in the same space as the training data to estimate the desired metrics and confidence intervals. We call this process the *serial approach* because each component in Figure 4.1 is designed and optimized serially. In the following sections we explain each component of the manifold regression framework shown in Figure 4.1.

Subsequent to this discussion, we present our two primary contributions to the problem of manifold-based metric estimation. First, due to the outliers issue discussed in Section 1.2.3, the embedding function obtained using the traditional Spectral Regression method cannot reconstruct the manifold built by Laplacian Eigenmaps accurately. Technically, the $L2$ norm in the SR objective function makes the embedding function sensitive to outliers. Therefore the Huber norm is employed to train the embedding function. This approach is called the *robust approach*. Second, the three metrics f_p, M_p and M_g are not independent, they follow the mathematical relation $f_p = M_p/(M_p + M_g)$. We enforce this

constraint into the training process to regularize the regression functions for the three metrics. We also determine the regression functions and embedding function simultaneously, thus this approach is called the *integrated approach*.

4.1 Laplacian Eigenmaps

As discussed in Section 3.1, the number of geometric features we compute is on the order of a few hundred. Dimensionality reduction is used to further extract from the data those degrees of freedom that are most relevant for solving the regression problem. As explained in the Section 1.2.3, our use of manifold methods is motivated by a desire to embed the training data comprised of the geometric feature vectors along with the known metrics into a low dimensional space where regression can be performed accurately. More precisely, we seek to transform the feature vectors into a space such that the distance between vectors in this new space is reflective of the distance between the corresponding source zone metrics we seek to determine. If this condition is satisfied, when the feature vector from a test datum is transformed into this space, the use of regression for the metrics based on the training data points close to the test data point in manifold space is expected to be accurate. As this closeness requirement involves a highly nonlinear mapping of the feature vectors [44], standard linear dimensionality reduction methods such as PCA [35] or LDA [17] discussed in the last chapter, are not appropriate.

In this section we use the Laplacian Eigenmaps (LE) approach to construct a manifold with the locality preserving property we desire. Mathematically, for each length $d + 3$ feature vector-metrics value pair $[\mathbf{x}_i, \mathbf{t}_i]^T$, we seek a low dimensional embedding, $\mathbf{r}_i \in \Re^m$

with $m \ll d+3$. Here $\mathbf{r}_i$ is best thought of as an m-dimensional coordinates vector for the i^{th} datum in the manifold. The LE manifold is constructed by choosing the manifold coordinates to minimize the following objective function [9],

$$\min_{\mathbf{r}_i \in \Re^m} \sum_{i,j=1}^{N} \|\mathbf{r}_i - \mathbf{r}_j\|_2^2 \, \omega_{ij} \tag{4.1}$$

The weight ω_{ij} is constructed as a measure of the similarity between $[\mathbf{x}_i, \mathbf{t}_i]^T$ and $[\mathbf{x}_j, \mathbf{t}_j]^T$, and is chosen to be largest when these quantities are closest. In this work we employ a variant of the Gaussian weight function [9] for which ω_{ij} is,

$$\omega_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma_1}\right) \exp\left(-\frac{\|\mathbf{t}_i - \mathbf{t}_j\|_2^2}{\sigma_2}\right) \tag{4.2}$$

we emphasize LE is used here to embed the three related metrics into the same manifold and we normalize all the metrics between 0 to 1.

To illustrate these ideas, consider again the three cases illustrated in Figures 4.2. With $\sigma_1 = 20$ and $\sigma_2 = 1$, the values used in our experiments in Section 4.6 when constructing the weight for comparing (a) and (b), the geometric feature vectors for these data are relatively similar (i.e., the Euclidean distance of feature vectors is 5.29, while the Euclidean distances of feature vectors between (a)/(c) and (b)/(c) are 7.43 and 7.49 respectively.), so that the first factor in (4.2) will be large, but the difference between the metric vectors is huge so that the second factor in (4.2) will be very small, resulting in $\omega_{ab} = 0.37$. When we calculate the weight for (a) and (c), the first factor measuring the similarity between geometric features will be relatively small, but the second factor will be large, yielding the weight $\omega_{ac} = 0.63$.

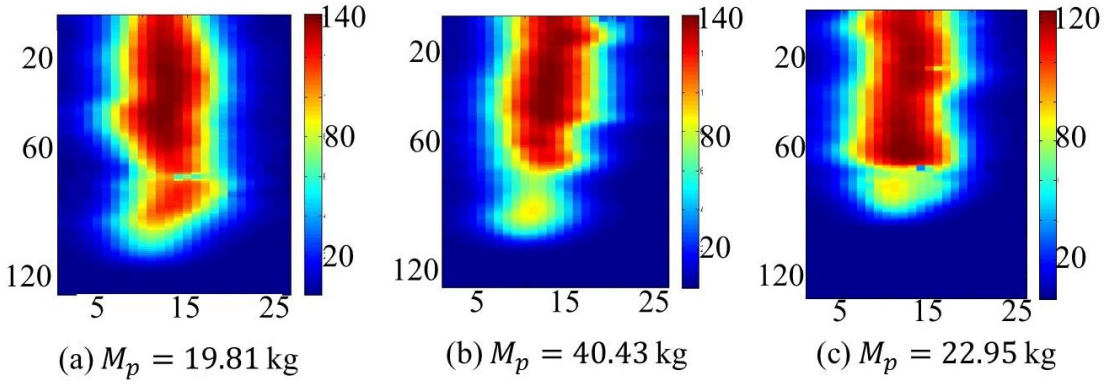

Figure 4.2: Ambiguity of the metric estimation problem. Due to the differences in the manner in which the contaminant was spilled into the subsurface, the pool masses associated with similar concentration images (a) and (b) are quite different while the dissimilar images (a) and (c) correspond to source zones with nearly the same mass in pools.

Thus, using this approach we see that (a) and (c) are, in a sense, almost twice as similar as (a) and (b) which is exactly what we desire given that metrics for (a) and (c) are much closer than those of (a) and (b).

In Figure 4.3, we display a two-dimensional projection of an $m = 4$ dimensional manifold constructed using this approach for the single metric mass in pools. As discussed in Section 2.4, we sort the eigenvalues of Laplacian matrix in ascending order, thus the minimum value of objective function (4.1) is $\sum_{k=1}^{m+1} \lambda_k$. The eigenvalues of Laplacian matrix are between 0 to 1 [9]. From Figure 4.4 we choose the eigenvalue from the second to the fifth and set $m = 4$ because the sixth eigenvalue is over 0.95. We show in Figure 4.3 the location of each training datum projected onto the first two coordinates corresponding to the smallest two eigenvalues. Each dot in this space corresponds to one datum of data set-4 used in the experiments in Section 4.6 with the color of the circle indicating the mass in pools, the colorbar indicates the range of mass in pools in kilogram. From Figure 4.3, the data whose metric values are similar will in fact be mapped close to one another at least in

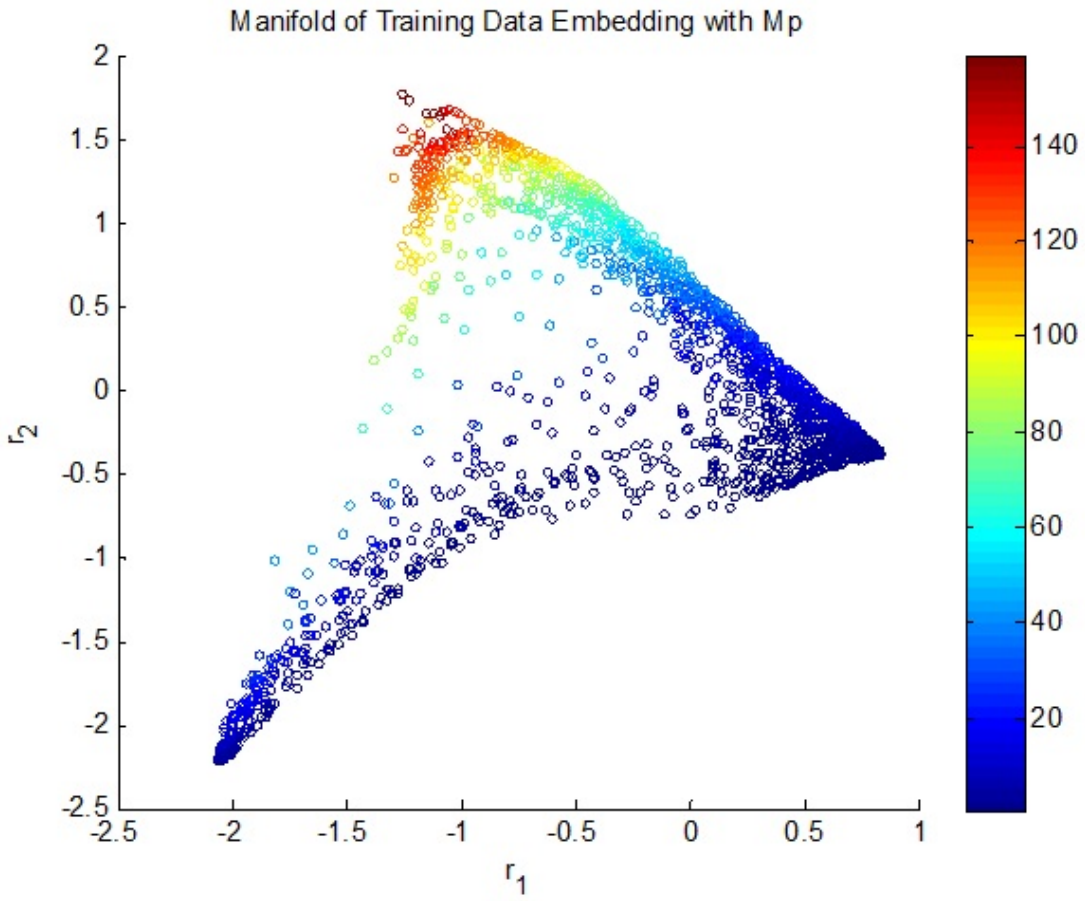

Figure 4.3: Embedding of concentration image data with its associated metric onto two dimensions, the color of dots indicates the mass in pools M_p as an example. The manifold we find has validated the objective of Laplacian Eigenmaps which is the data with metric locate nearby each other, this gives advantage for linear regression function learning.

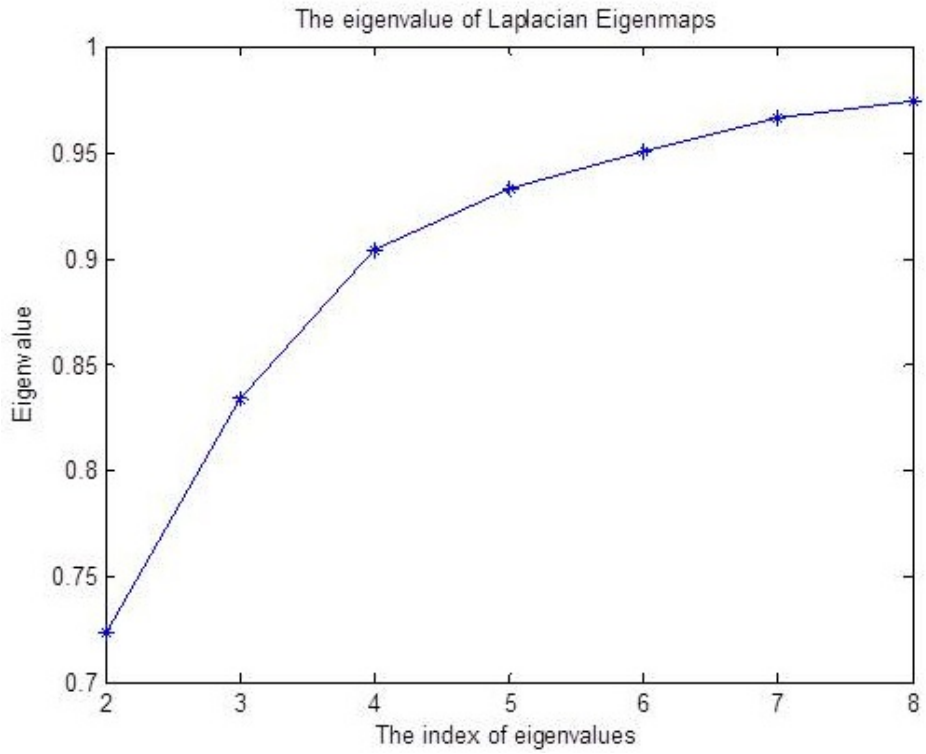

Figure 4.4: The plot of second to eighth eigenvalue of Laplacian matrix for LE.

this subspace so that the locality preserving property discussed before is in fact evident at least with respect to the training data.

Two remarks are in order concerning the LE method. First, the process of solving the optimization problem in (4.2) is well known in Belkin *et al.*'s work [9]. The details are reviewed in Section 2.4. Second, full specification of the problem requires that the hyper-parameters σ_1 and σ_2 be provided. As described in the Section 4.6.2, here we use a cross validation approach [11] to determine these quantities adaptively from the data sets.

4.2 Spectral Regression

After LE embeds the training data in manifold space, the predictions of metrics given a test datum are processed by first constructing geometric feature vector from observed test concentration image and then embedding this feature vector into the manifold. Finally the estimation of the associated metrics is obtained as a linear combination of the metric vectors associated with training data that are located close-by in the manifold. Now, the manifold is constructed under the assumption that both the $\mathbf{x}_i$ and $\mathbf{t}_i, i = 1, \cdots, N$ are known. Thus, a method for embedding data when only $\mathbf{x}_i$'s are given is needed. In Section 2.5, we discussed the limitation of Nyström formula and in this section the Spectral Regression method is used to perform the task of learning a function $\mathbf{r} = f(\mathbf{x})$ which can embed the geometric feature of test data into the manifold space.

Spectral Regression (SR) casts embedding function learning into a regression framework. Since LE is a nonlinear dimension reduction method, the mapping function from feature space to manifold space should also be nonlinear. We assume the nonlinear function lives in a Hilbert space specified by a kernel $k(\cdot, \cdot)$ [64]. The model of this nonlinear function is $f(\mathbf{x}) = \mathbf{A}^T\mathbf{k}(\mathbf{x})$, specified by the parameter matrix $\mathbf{A} \in \Re^{N \times m}$ and kernel function vector $\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), k(\mathbf{x}, \mathbf{x}_2), \cdots, k(\mathbf{x}, \mathbf{x}_N)]^T$. The optimization problem to learn $\mathbf{A}$ is [12]:

$$\mathbf{A}^* = \underset{\mathbf{A}}{\operatorname{argmin}} = \sum_{i=1}^{N} \|\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) - \mathbf{r}_i\|_2^2 + \gamma\|\mathbf{A}\|_F^2 \tag{4.3}$$

where γ is a regularization hyper-parameter. The solution of problem (4.3) is $\mathbf{A}^* = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{R}^T$ where matrix $\mathbf{K}$ is the Gram matrix with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N]$.

In the experiments Section 4.6, we use the Gaussian kernel [12],

$$k(\mathbf{x}, \mathbf{x}_i) = \exp(-\frac{\|\mathbf{x} - \mathbf{x}_i\|_2^2}{\sigma_{SR}}) \tag{4.4}$$

where the hyper-parameter σ_{SR} is determined by cross validation.

4.3 Bayesian Regression

The final step of our machine learning framework is training the regression function in manifold space to estimate the metrics of test data. One of the advantages of Bayesian regression is that it can provide both the estimated metric itself along with a confidence interval allowing for the quantification of uncertainty in the estimate. The details of standard linear Bayesian regression is provided in Bishop's book [11]. Here we summarize our approach.

We first centralize the manifold coordinate matrix and metrics vectors, that is, we compute $\bar{\mathbf{r}} = \sum_{i=1}^{N} \mathbf{r}_i/N$, and define a centralized $\mathbf{R}$ as $\bar{\mathbf{R}} = [\mathbf{r}_1 - \bar{\mathbf{r}}, \mathbf{r}_2 - \bar{\mathbf{r}}, \cdots, \mathbf{r}_N - \bar{\mathbf{r}}]$, and centralize $\mathbf{T}$ as $\bar{\mathbf{T}} = [\mathbf{t}_1 - \bar{\mathbf{t}}, \mathbf{t}_2 - \bar{\mathbf{t}}, \cdots, \mathbf{t}_N - \bar{\mathbf{t}}]$ where $\bar{\mathbf{t}} = [\bar{f_p}, \bar{M_p}, \bar{M_g}]^T = \sum_{i=1}^{N} \mathbf{t}_i/N$. A linear regression function is used to estimate each metric which takes the form $\bar{t}(\bar{\mathbf{r}}_i) = \mathbf{w}^T \bar{\mathbf{r}}_i + \epsilon$, where ϵ is modeled as zero mean, additive Gaussian noise with variance β. We employ an iterative maximum-likelihood method to estimate β [50]. Specifically all experiments, we initially set $\beta = 0.01$. After determining $\mathbf{w}^*$, the optimal values for $\mathbf{w}$ using the methods of [11], the random variable ϵ_i is estimated as $\hat{\epsilon}_i = \bar{t}_i - \mathbf{w}^{*T} \bar{\mathbf{r}}_i$, and β is updated as the sample variance of $\hat{\epsilon}_i$. With this new value of β, the regression function can again be computed and the process repeats until convergence. The rate of convergence is quite fast, generally only

one update is enough. The complete Bayesian regression method is provided in Section 2.3.

For the fully Bayesian approach to regression construction [11], we also require a prior probability distribution for the unknown metrics being estimated for a given set of test data. Again a zero mean Gaussian model is used for which the variance needs to be determined. Unlike the specification of β here we only possess the metrics estimated from a single set of test data so that sample variance methods can not so easily be employed. To address this issue we assume that the embedding process is sufficiently accurate so that the training data in the neighborhood of the embedded test data will have associated metrics that are close to that of the test data. Under such an assumption, we estimated the variance of the metric for the test datum as the variance of the metrics within the n-nearest-neighbors of test datum in the manifold. The size of neighborhood, n, is determined empirically to ensure that roughly 85% of the test data did in fact fall in the theoretical 85% confidence interval. From our experiments using an integrated approach with half of data set-4 for training, the EP85's of size 10 for all the metrics are around 85%. Therefore we choose the size n as 10.

4.4 Robust Spectral Regression

One approach to using the methods described in the last sections to address the problem of interest here is to optimize each of the components shown in Figure 4.1 individually. More specifically, the Laplacian Eigenmpas method could be used to construct a manifold in which the training data corresponding to similar metric vectors are, ideally, located close to one another. Subsequently, the same data set with only geometric feature $\mathbf{x}$ would be

used to determine the optimal $\mathbf{A}$ matrix for the spectral regression function by minimizing the reconstruction error $e = \|\mathbf{A}^T\mathbf{k}(\mathbf{x}) - \mathbf{r}\|_2$. Finally, the training data would be used to construct the Bayesian regression functions. The performance of the overall processing chain could then be evaluated using the test data.

Unfortunately, this serialized type of approach suffers from challenges associated with the embedding of the feature vector into the constructed manifold. To illustrate the more general situation, consider the simpler problem of estimating only the mass in pools. In Figure 4.5(a) and (c) we show the manifold determined by LE and the reconstructed manifold obtained using SR. Ideally the two would be the same. In general we would hope that the data points predicted by SR are close to their "true" coordinates as defined by the LE process. By comparing these two figures however we see that there is a good deal of discrepancy.

In a bit more detail, recall that the goal of SR is to determine a function for embedding the geometric data vectors into the manifold when the associated metric values are not known. The hope is that such an embedding function will place the test data into the manifold close to the training data coordinates such that we can estimate the metrics vector using Bayesian regression based on the neighbors. Unfortunately, in Section 1.2.3 around the discussion of Figure 1.5, the metric values associated with very different concentration data sets can be quite similar leading to large errors in the embedding process. The concentration images of Figure 1.5 are shown in Figure 4.6 for convenience.

By zooming in the red rectangle in Figure 4.5(a) we can better see where these data are located in the manifold. Figure 4.5(b) indicates that the cases associated with Figure 4.6(b) and (c) are placed in the low M_p neighborhood, although the case for Figure 4.6(a) is a

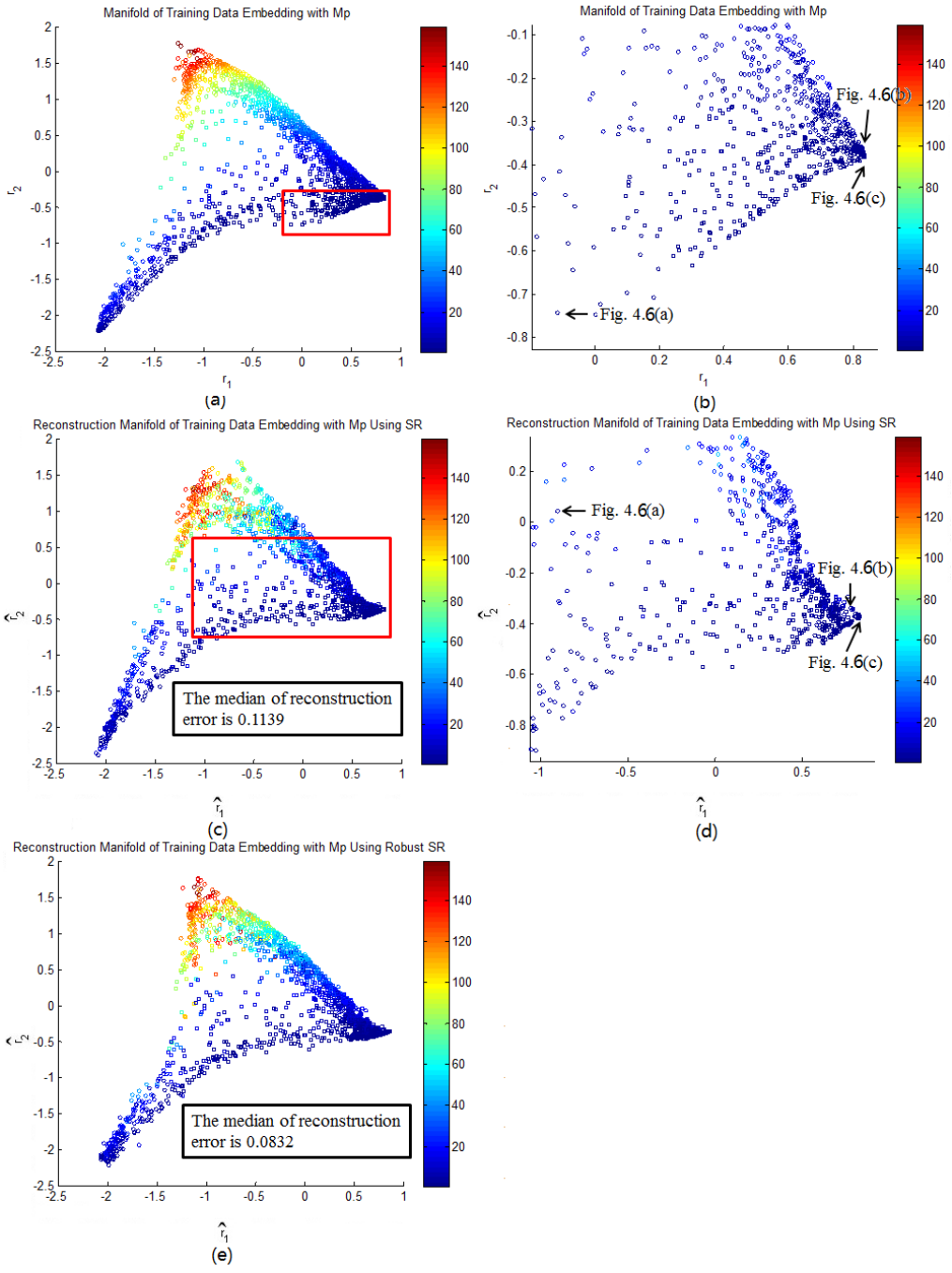

Figure 4.5: The manifold constructed by LE using feature and metrics pair is shown in (a), the manifold coordinates of data for Figure 4.6 is in (b), the reconstructed manifold using embedding function with only geometric feature is shown in (c), the reconstructed manifold coordinates of data for Figure 4.6 is in (d) and the reconstructed manifold by embedding function from Robust SR is shown in (e). The median of the reconstructed error is in the legends in both (c) and (e).

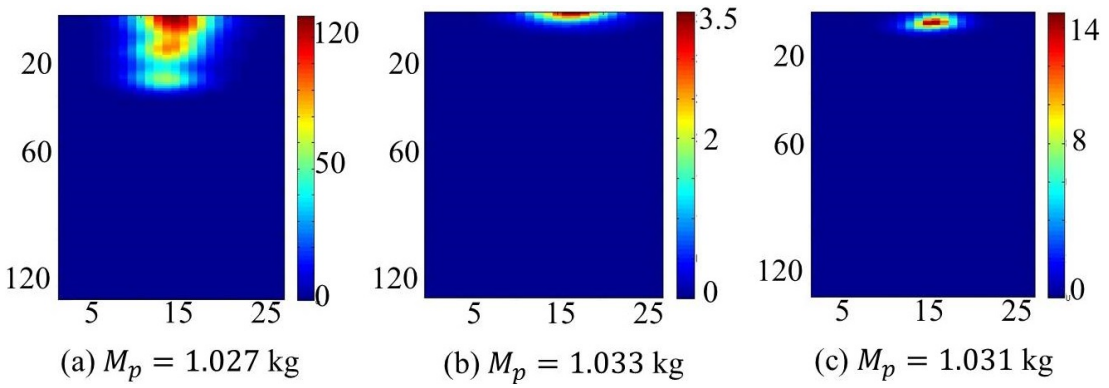

Figure 4.6: The outlier in data set, (a)-(c) have almost the same metric mass in pools, but due to the different spill scenario, the concentration data is quite different.

bit further away from (b) and (c) due to difference in the geometric feature vector. After SR, we can reconstruct the manifold found by LE using only geometric feature vector as $\hat{\mathbf{r}} = \mathbf{A}^T\mathbf{k}(\mathbf{x})$, which is shown in Figure 4.5(c). By zooming in the red rectangle in Figure 4.5(c), Figure 4.5(d) shows that concentration (b) and (c) are reconstructed almost at the same position as in Figure 4.5(b), but the embedding of (a) is quite far from where it should be. Moreover, in the neighborhood of this point the associated training data have high M_p value as indicated by the bluish color. In Table 4.1, we summarize the quantitative analysis of concentration data in Figure 4.6. From Table 4.1, the reconstructed error (defined to be $\|\hat{\mathbf{r}} - \mathbf{r}\|_2$) of case (a) is much larger than (b) and (c).

Concentration	Manifold Coordinate in Figure. 4.5(b)	Reconstructed Manifold Coordinate in Figure. 4.5(d)	Reconstruction Error
(a)	(-0.2029,-0.7378)	(-0.8092,0.0455)	0.9905
(b)	(0.8320,-0.3802)	(0.7759,-0.3940)	0.0577
(c)	(0.8201,-0.3621)	(0.8081,-0.3601)	0.0121

Table 4.1: The quantitative analysis of the data in Figure 4.6

Since cases like Figure 4.6(a) are relatively rare, we regard them as the outliers. In order to get an embedding function that is less sensitive to these potentially large errors, we

propose in this section a robust form of the spectral regression method. In the original SR objective function (4.3), the quadratic function will penalize heavily the large reconstruction error, which makes the embedding function quite sensitive to outliers. The Huber norm which is proposed by [29] can solve this problem and is defined as

$$H(x) = \begin{cases} \frac{x^2}{2\varepsilon} & \text{if } x < \varepsilon \\ x - \frac{\varepsilon}{2} & \text{if } x > \varepsilon \end{cases} \tag{4.5}$$

The threshold ε distinguishes normal data from outliers. To determine this threshold we first use (4.3) to determine an embedding function and then measure the reconstructed error, $e_i = \|\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) - \mathbf{r}_i\|_2,\ i = 1, \cdots, N$. We let ε be the 95th percentile of $\{e_i\}$ [50]. Using this value (4.3) is changed to the following as the basis for determining the optimal embedding function:

$$\mathbf{A}^* = \underset{\mathbf{A}}{\operatorname{argmin}} = \sum_{i=1}^{N} H(\|\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) - \mathbf{r}_i\|_2) + \gamma\|\mathbf{A}\|_F^2 \tag{4.6}$$

In the case of pool mass, the reconstructed manifold using robust SR is shown in Figure 4.5(e). We see that the shape of reconstructed manifold is improved by visually comparing this figure to Figure 4.5(c). The median of reconstruction error of Figure 4.5(c) is 0.1139, while the median of that for Figure 4.5(e) drops to 0.0832. Since we reduce the impact of outliers on the learning the embedding function, the outliers cannot be embedded correctly in the manifold using the robust approach, but by using the Huber norm in the robust cost function, the impact of these outliers on the overall accuracy of the SR embedding is reduced.

4.5 Integrated Approach

Thus far, we have considered the estimation of these three quantities f_p, M_p, M_g individually. That is, one learner is used to estimate pool fraction, another for the mass of DNAPL in pools and a third for the mass of DNAPL in ganglia. These three metrics however are not independent of one another. Thus, we hypothesize that an approach which incorporates the mathematical relationships among these quantities to determine all three at once should outperform the case where we ignore the coupling. More specifically, we exploit the fact that pool fraction is equivalent to the ratio of the mass in pools to the mass in pools plus the mass in ganglia.

The analytical method we have developed proceeds as follows. We first use LE to determine the manifold coordinates of the training data $\mathbf{R}$, and then simultaneously determine three Bayesian regression functions, one each for the three metrics of interest, integrated with robust SR, subject to the constraint that the estimated pool fraction is equal to the ratio of M_p to $M_p + M_g$. The optimization problem to solve then is,

$$\begin{aligned} \min_{\mathbf{A},\mathbf{W}} \quad & L(\mathbf{A},\mathbf{W}) = \frac{1}{N}\sum_{i=1}^{N} H(\|\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) - \mathbf{r}_i\|_2) + \gamma_1\|\mathbf{A}\|_F^2 + \\ & \gamma_2 \frac{1}{N}\sum_{i=1}^{N} H(\|\mathbf{W}^T\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) - \mathbf{t}_i\|_2) + \gamma_3\|\mathbf{W}\|_2^2 \\ \text{s.t.} \quad & \mathbf{w}_1\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) + \bar{f}_p = \frac{\mathbf{w}_2\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) + \bar{M}_p}{\mathbf{w}_2\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) + \bar{M}_p + \mathbf{w}_3\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) + \bar{M}_g} \quad i = 1,\ldots,N. \end{aligned} \tag{4.7}$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3]$ and the estimation of pool fraction, mass of DNAPL in pool and

ganglia are generated using the following linear regression functions.

$$\begin{aligned}\hat{f}_p &= \mathbf{w}_1^T \mathbf{A}^T \mathbf{k}(\mathbf{x}_i) + \bar{f}_p \\ \hat{M}_p &= \mathbf{w}_2^T \mathbf{A}^T \mathbf{k}(\mathbf{x}_i) + \bar{M}_p \\ \hat{M}_g &= \mathbf{w}_3^T \mathbf{A}^T \mathbf{k}(\mathbf{x}_i) + \bar{M}_g\end{aligned} \tag{4.8}$$

The integrated robust SR and linear Bayesian regression is proposed in the objective function (4.7). The constraint in (4.7) enforces the physical relationship among the three quantities to be estimated. The first term in (4.7) is motivated by the robust SR cost function with the Huber norm. The second term $\|A\|_F^2$ is the regularization term coming from SR, with the regularization hyper-parameter γ_1 playing the same role as γ in (4.6). The third term in (4.7) arises from the mathematical details of the Bayesian regression problem provided in [11]. The hyper-parameter γ_2 is used to balance the desire for a good embedding with the needs of obtaining accurate regression results. Using Lagrange multiplier, we can convert (4.7) into unconstrained optimization problem as the following,

$$\begin{aligned}\min_{\mathbf{A},\mathbf{W}} \quad & L(\mathbf{A},\mathbf{W}) = \frac{1}{N}\sum_{i=1}^{N} H(\|\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) - \mathbf{r}_i\|_2) + \gamma_1\|\mathbf{A}\|_F^2 + \\ & \gamma_2 \frac{1}{N}\sum_{i=1}^{N} H(\|\mathbf{W}^T\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) - \mathbf{t}_i\|_2) + \gamma_3\|\mathbf{W}\|_2^2 + \\ & \gamma_P \sum_{i=1}^{N}\left(\mathbf{w}_1\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) + \bar{f}_p - \frac{\mathbf{w}_2\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) + \bar{M}_p}{\mathbf{w}_2\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) + \bar{M}_p + \mathbf{w}_3\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) + \bar{M}_g}\right)^2.\end{aligned} \tag{4.9}$$

While (4.9) provides for the joint determination of all three regression functions in a manner that reflects the physical relationship among the metrics. To solve problem (4.9) we use a cyclic decent type of optimization algorithm which is summarized in the Table 4.2.

The details about the gradient are provided in Appendix A. For $\xi_0 = \xi_1 = 0.001$ the decent algorithm converged for all experiments in this thesis.

Algorithm: Cyclic Decent Algorithm for Integrated Approach

Inputs: $(\mathbf{R}, \mathbf{T}) := \{\mathbf{r}_i, \mathbf{t}_i\}_{i=1}^{N}, \eta, \sigma_1, \sigma_2, \sigma_{SR}, \gamma_1, \gamma_2, \gamma_3, \gamma_P, \xi_0 > 0$.
Outputs: Embedding function $\mathbf{A}$ and weight vectors of regression functions $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$.
Initialization: $\mathbf{A}^{(0)}$ and $\mathbf{w}_1^{(0)}, \mathbf{w}_2^{(0)}, \mathbf{w}_3^{(0)}$ using serial approach.
Repeat:
- Gradient Descent Update $\mathbf{A}^{(p)}$ and $q = 1$.
  - Compute $\mathbf{A}_{q+1}^{(p+1)} = \mathbf{A}_q^{(p)} - \xi_q \frac{\partial L}{\partial \mathbf{A}_q^{(p)}}$.
  - $q = q + 1, \xi_q = \frac{\xi_0}{q}$.
  - **Until** $\|\mathbf{A}_{q+1}^{(p+1)} - \mathbf{A}_q^{(p+1)}\|_2^2 < \eta$.
- Gradient Descent Update $\mathbf{w}_k^{(p)}$ and $q = 1, k = 1, 2, 3$.
  - Compute $\mathbf{w}_{k,q+1}^{(p+1)} = \mathbf{w}_{k,q}^{(p)} - \xi_q \frac{\partial L}{\partial \mathbf{w}_{k,q}^{(p)}}$.
  - $q = q + 1, \xi_q = \frac{\xi_0}{q}$.
  - **Until** $\|\mathbf{w}_{k,q+1}^{(p+1)} - \mathbf{w}_{k,q}^{(p+1)}\|_2^2 < \eta$.

Until: $\|\mathbf{w}_k^{(p+1)} - \mathbf{w}_k^{(p)}\|_2^2 < \eta, k = 1, 2, 3$.

Table 4.2: The cyclic decent algorithm to solve integrated approach.

4.6 Experiments

We evaluate the performances of three approaches in this section. Under the serial approach, each module in Figure 4.1 is determined independently using the ideas discussed from Section 4.1 to 4.3. We compare this serial approach with the new approaches proposed in Section 4.4 and 4.5. To quantify the utility of our robust approach to SR, we change the objective function of SR from (4.3) to (4.6) keeping the remaining components the same as the serial approach. Finally, we evaluate the fully integrated approach where the objective function is (4.9), the embedding function and three manifold regression functions are determined simultaneously. We compare the performances of these approaches to verify

that the integrated approach can improve the accuracy of the metrics estimation.

4.6.1 SGS Data Sets

In this section, we use four data sets to test the performance of our manifold regression approaches. The hydraulic conductivity of these data is generated by Sequential Gaussian Simulation [22], here we refer our data sets used for experiments as SGS data sets. The parameter settings of hydraulic conductivity field for SGS data sets were introduced in Section 2.6.1. In data set-1, the hydraulic conductivity is a relatively homogeneous glacial out-wash deposit, it combines three different spill scenarios which are summarized in Table 2.1. In data set-2 and 3, the statistical parameters of hydraulic conductivity are changed to larger correlation length values and higher lognormal transformed hydraulic conductivity variance σ^2 $(ln(k))$ of 1.0 and 1.5 respectively while keeping the same spill scenario. The sizes of data set-1 to 3 are 500, 600 and 900 respectively. The conditions used in data set 2 and 3 generated DNAPL architectures with higher pool content and longer sustained dissolution in time compared to data set 1. Since the down-gradient dissolved concentration was sampled every 20 time steps for all the data sets, those with longer dissolution times generated more data samples (data sets 2 and 3 with 600 and 900 samples, respectively) than those with less persistent dissolution times (data set 1 with 500 samples). In order to test the performance of regression functions under a wide range of conditions, we combine these three data sets as data set-4.

4.6.2 Hyper-parameters Selection

The choice of hyper-parameters σ_1 and σ_2 in (4.2) for LE and σ_{SR} in (4.4) for the SR kernel function controls the embedding of data in manifold space. The choice of regularization parameters γ_1, γ_2 and γ_3 in (4.9) will affect the performance of regression functions and γ_P controls the importance of enforcing the physical relationship among three metrics. In practice, we need to determine the values of these hyper-parameters before we run our algorithm. Here we use cross validation [11] to accomplish this task. As an example of the results of this process, the hyper-parameter settings of all SGS data sets are determined in the following way. A grid search [33] process is used with data set-4 because it is the most general data set which combines the data generated under different conditions. For σ_1, σ_2 and σ_{SR}, we choose five candidates for each hyper-parameter. For each combination of σ_1, σ_2 and σ_{SR}, cross validation is used to evaluate the empirical performance of regression by serial approach. In a bit more detail, we randomly select 90% of data set for training and the remaining 10% for testing, repeating this procedure for 10 times. The final values of σ_1, σ_2 and σ_{SR} are chosen as those with the smallest regression error for the sum of all three metrics. Then these selected σ's are applied to data set-1 to 4. The grid search results are listed in Table 4.3. In the table, sum of the median absolute error between the true metric and the estimated metric for each combination of σ's are shown. In the Table 4.3, each row we keep σ_1 fixed, each column we keep σ_2 fixed and the values of σ_{SR} are listed in the first column of table. From the experimental results, we select $\sigma_1 = 20, \sigma_2 = 1$ and $\sigma_{SR} = 15$ because the estimations of three metrics reach the sum of their smallest regression error with this combination. Using these hyper-parameters, cross validation is then employed to

determine the values of the hyper-parameters $\gamma_1, \gamma_2, \gamma_3$ and γ_P for integrated approach. In our experiment $\gamma_1 = 0.5, \gamma_2 = 1000, \gamma_2 = 25, \gamma_P = 2$. For the Bayesian regression, the noise in the original data set is small, so we set the initial variance of metrics β as 0.01 and for the prior distribution of $\mathbf{w}$, we set $\alpha = 1$.

	σ_1	σ_2				
		0.1	0.5	1	2	5
$\sigma_{SR} = 5$	10	16.73	13.04	10.90	13.17	18.16
	15	13.43	11.81	10.61	12.21	13.86
	20	12.30	11.61	9.57	11.28	12.28
	25	14.37	12.17	10.10	12.33	12.65
	30	13.89	12.80	12.09	12.22	12.91
$\sigma_{SR} = 10$	10	18.16	13.13	11.29	14.44	16.52
	15	12.75	11.98	10.73	12.11	13.35
	20	12.33	11.02	9.64	10.74	12.00
	25	12.91	11.93	11.65	12.14	12.25
	30	14.20	12.96	10.56	12.53	13.86
$\sigma_{SR} = 15$	10	16.93	12.66	10.77	13.16	16.76
	15	12.40	11.81	10.51	11.21	12.62
	20	12.63	11.32	**9.44**	10.84	12.53
	25	13.91	12.05	9.68	11.73	12.10
	30	14.30	12.80	10.07	11.92	13.51
$\sigma_{SR} = 20$	10	19.42	12.44	10.93	13.43	15.45
	15	13.07	11.72	10.49	11.76	13.16
	20	12.54	10.84	9.82	11.00	12.62
	25	14.02	12.83	10.00	11.80	12.58
	30	14.65	12.94	10.15	12.80	12.88
$\sigma_{SR} = 25$	10	18.43	12.41	11.00	13.50	16.36
	15	12.63	11.88	11.12	11.93	14.45
	20	12.25	11.59	9.75	10.88	12.47
	25	13.82	12.68	10.02	12.01	12.57
	30	15.32	13.65	10.40	12.18	12.89

Table 4.3: The hyper-parameters selection of σ_1, σ_2 and σ_{SR} using data set-4.

4.6.3 Experimental Results

In our experiments, we evaluate the performance of our approaches in the following way. For data sets one through three, we explored the performance of our processing methods

using 10%, 20%, $\cdots$, 90% of the data for training and the balance for testing. As long as more than about 30% to 50% of the data (depending on the metric) were used for training, the results were largely the same. Thus here we consider the case where 90% of data are employed for training and the remaining 10% for testing. In the case of data set-4, which combines all of the data from the other three sets, there are 2000 samples. Now, when training data are plentiful, there is little to be gained by employing the more complex processing method in which the physical constraint among the three metrics is explicitly enforced. Therefore we decided to randomly select half of data set for training and use the other half for testing. The procedure is then repeated 10 times. To demonstrate the utility of the physical constraint in data-poor scenario, we also consider the case where only 25% of data set-4 is used for training.

The geometric feature we use for manifold regression includes the number of connected component and the percentage of remaining area. Both are predictive for the metrics estimation. When we use only the number of connected component as the feature, the sum of median error for all metrics is 11.78; and when we only take the percentage of remaining area as the feature, the sum of median error for all metrics is 11.81. From Table 4.3, the sum of median error is 9.44 when the both of features are applied.

In order to show that the performance of our manifold regression framework is better than classification framework, we calculate the mean absolute error *mean* ε_a where $\varepsilon_a = |t - \hat{t}|$. The mean ε_a is the average error over whole test data set. Comparing the precision between classification and our serial approach results using data set-1[1], for our application, we find the mean ε_a of pool fraction is 0.102 which is smaller than the width of the narrowest

[1]We only have data set-1 when we first propose classification framework.

interval (i.e., 0.200) in classification; the mean ε_a of mass in pools is 5.24 kg which is less than the width of the narrowest interval 8.67 kg and the mean ε_a of mass in ganglia is 12.4 kg which is less than 13.4 kg in classification. These demonstrate the efficacy of our manifold regression framework.

In order to show the superior performance of our integrated approach, the statistical results for comparison between serial approach, robust approach and integrated approach are provided in Table 4.4 to 4.8. The range of metrics for the each data set are also provided in these tables, $\varepsilon_r = \varepsilon_a/t \times 100\%$ is the relative error. We also provide the Empirical Percentage of true metrics falling into the 85% confidence intervals, denoted as *EP85*. Indeed, the Bayesian regression implies a Gaussian distribution $\mathcal{N}(\hat{t}, \hat{s}^2)$ for each metric estimation, we take the mean as the estimation and $\hat{s}^2$ is the variance [11], thus the 85% confidence interval is $[\hat{t} - 1.44\hat{s}, \hat{t} + 1.44\hat{s}]$.

In Table 4.4, the results of estimating the three metrics using data set-1 are shown. This is the smallest data set we used. From the statistical results, (e.g. the median error of M_p drops from 3.15 kg to 1.97 kg), enforcing the physical relation between metrics indeed improves the performance of regression functions when the small training data set is applied. In Table 4.5 and Table 4.6, the statistical results of f_p for data sets-2 and 3 are both better than those of data set-1. We believe that the reason is these two data sets are generated under one spill scenario. The range of metric M_p for data set-2 and 3 are much higher than that for data set-1, and while the relative error is smaller than that of data set-1, the median value is a bit high. From the definition of relative error, we have the true metric is $t = \varepsilon_a/\varepsilon_r \times 100\%$, therefore in the case where ε_a is large and ε_r is small, t is large, this high absolute error is less of a problem. Since most DNAPL is entrapped in pools, the range

of metric M_g for data set-2 and 3 is smaller than data set-1, thus the median error is also small.

Data set-4 combines all SGS data generated under a wide range of conditions. In Table 4.7, the median error of M_p for the integrated approach is around 4 kilograms and that of M_g is less than 1.5 kg. Indeed, ganglia mass here is relatively easy to determine because the morphology of the concentration images changes significantly when ganglia are present. From the result of pool fraction in Table 4.7, the median absolute error level is around 0.05 (meaning half the time we are within 0.05 of the true pool fraction). When measured relative to the true pool fraction, the median level is around 8%.

From Table 4.7, the estimation of f_p is very accurate, the median ε_a and median ε_r of integrated approach are both very small. But the relative errors of mass in pools and mass in ganglia are worse than that of pool fraction, thus in Figure 4.7, we show the scatter plots of absolute error versus relative error. In Figure 4.7(a), the scatter plot of errors for mass in pools shows that the data with high relative error have small absolute error while the data with high absolute error have small relative error. By the definition of ε_r, the true values of test data with high ε_r and low ε_a are small. A little overestimation of $\hat{M}_p$ for a small true value of M_p is not a big problem. There are some test data in Figure 4.7(a) that both ε_r and ε_a are large, we believe they are the outliers because our robust spectral regression are not sensitive to outliers, thus it can not locate the outlier test data in the right place in the manifold. The same explanation can be applied to mass in ganglia.

In order to further test the performance of our proposed integrated approach, we reduce the size of training data to 25%. From Table 4.8, the accuracy of manifold regression function with the robustness and physics constraint under small training data set is almost

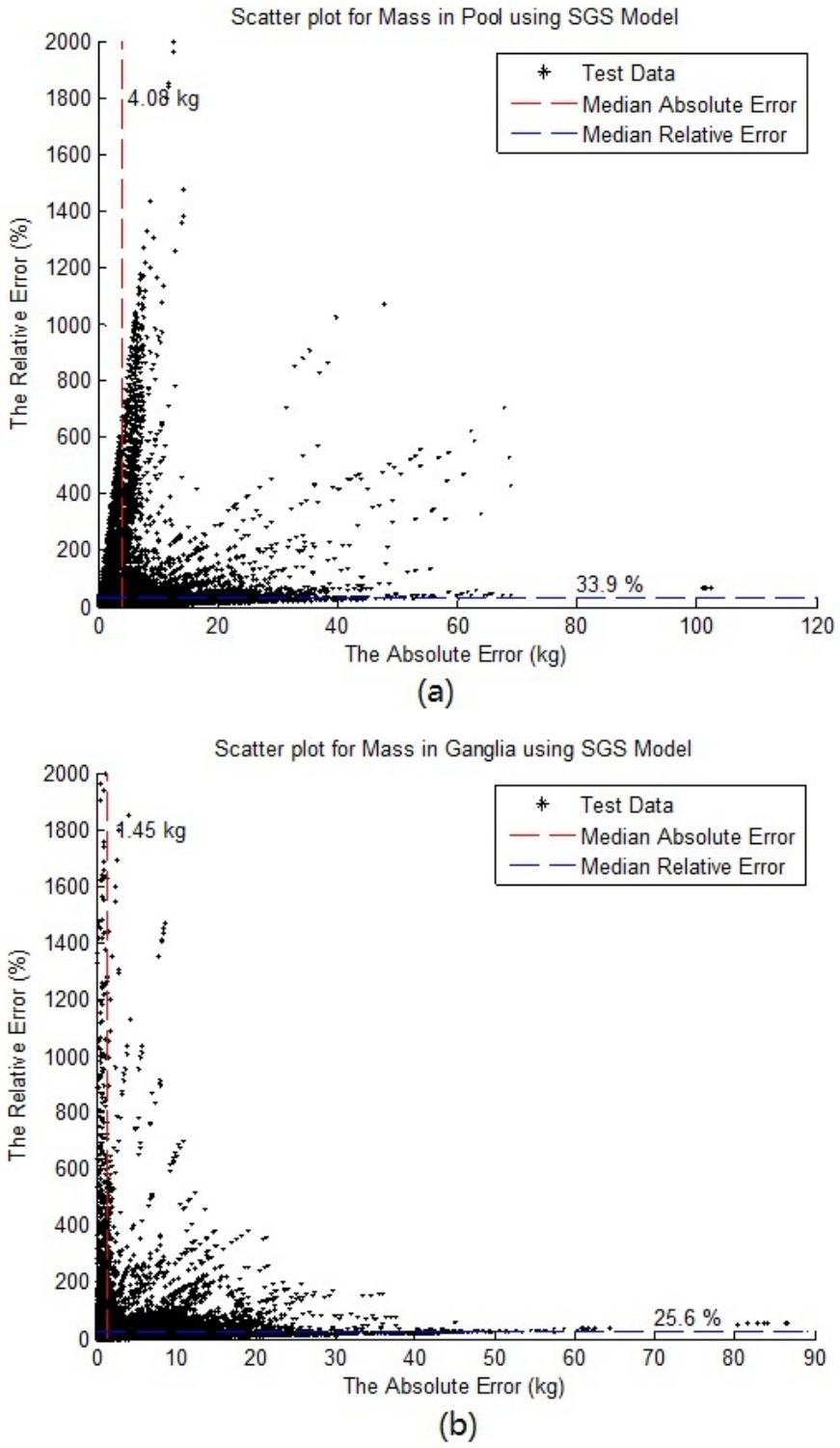

Figure 4.7: The scatter plot of absolute error and relative error. Each asterisk in the plots represent one test datum. The x-coordinate is absolute error and y-coordinate is relatice error. Sub-figure (a) shows the mass in pools and sub-figure (b) shows the mass in ganglia.

	The Range of Metric	The Approaches	median ε_a	median ε_r	EP85
f_p	$0 \sim 1$	Serial Approach	0.0712	34.90%	85.60%
		Robust Approach	0.0606	32.30%	91.60%
		Integrated Approach	0.0451	27.10%	93.60%
M_p	$0.59 \sim 44.62$ (kg)	Serial Approach	3.51	50.30%	88.60%
		Robust Approach	2.24	42.30%	84.60%
		Integrated Approach	1.97	36.90%	86.20%
M_g	$0 \sim 201.4$ (kg)	Serial Approach	7.39	22.40%	86.00%
		Robust Approach	5.46	16.10%	91.00%
		Integrated Approach	4.92	14.10%	91.20%

Table 4.4: The statistical results and Empirical Percentage (EP85) of data set-1.

	The Range of Metric	The Approaches	median ε_a	median ε_r	EP85
f_p	$0 \sim 1$	Serial Approach	0.0457	5.99%	93.80%
		Robust Approach	0.0408	5.06%	92.17%
		Integrated Approach	0.0389	4.81%	85.33%
M_p	$0.61 \sim 139.5$ (kg)	Serial Approach	3.73	36.70%	86.60%
		Robust Approach	3.18	28.80%	90.67%
		Integrated Approach	2.14	24.80%	91.33%
M_g	$0 \sim 114.4$ (kg)	Serial Approach	1.52	29.20%	85.80%
		Robust Approach	1.33	20.50%	85.83%
		Integrated Approach	0.896	19.70%	89.50%

Table 4.5: The statistical results and Empirical Percentage (EP85) of data set-2.

	The Range of Metric	The Approaches	median ε_a	median ε_r	EP85
f_p	$0 \sim 1$	Serial Approach	0.0443	5.21%	85.11%
		Robust Approach	0.0362	4.26%	92.67%
		Integrated Approach	0.0333	3.90%	84.33%
M_p	$0.60 \sim 159.7$ (kg)	Serial Approach	5.19	29.60%	86.44%
		Robust Approach	3.33	19.70%	85.44%
		Integrated Approach	2.57	14.10%	85.11%
M_g	$0 \sim 91.6$ (kg)	Serial Approach	1.26	37.10%	94.89%
		Robust Approach	0.978	29.80%	87.78%
		Integrated Approach	0.637	16.50%	88.11%

Table 4.6: The statistical results and Empirical Percentage (EP85) of data set-3.

	The Range of Metric	The Approaches	median ε_a	median ε_r	EP85
f_p	$0 \sim 1$	Serial Approach	0.0622	9.52%	90.70%
		Robust Approach	0.0561	8.15%	88.45%
		Integrated Approach	0.0505	8.07%	88.53%
M_p	$0.59 \sim 159.7$ (kg)	Serial Approach	5.41	43.40%	85.33%
		Robust Approach	4.75	38.20%	85.33%
		Integrated Approach	4.08	33.90%	85.87%
M_g	$0 \sim 201.4$ (kg)	Serial Approach	3.97	47.10%	94.80%
		Robust Approach	1.75	28.90%	85.16%
		Integrated Approach	1.45	25.60%	85.13%

Table 4.7: The statistical results and Empirical Percentage (EP85) of data set-4 using half of dataset for training.

	The Range of Metric	The Approaches	median ε_a	median ε_r	EP85
f_p	$0 \sim 1$	Serial Approach	0.0822	11.40%	93.17%
		Robust Approach	0.0679	10.50%	92.55%
		Integrated Approach	0.0533	8.41%	88.19%
M_p	$0.59 \sim 159.7$ (kg)	Serial Approach	6.02	49.10%	89.23%
		Robust Approach	5.26	39.30%	84.40%
		Integrated Approach	4.63	37.60%	86.50%
M_g	$0 \sim 201.4$ (kg)	Serial Approach	4.34	60.80%	94.97%
		Robust Approach	2.68	42.30%	85.88%
		Integrated Approach	1.84	29.10%	85.19%

Table 4.8: The statistical results and Empirical Percentage (EP85) of data set-4 using a quarter of dataset for training.

that same as that under large training data set, and has a significant improvement comparing to the other two approaches. To gain a better understanding of the performance of the approaches we proposed in this chapter, in Figure 4.8 we compare the 10th to 90th percentile [50] of absolute error for all three approaches using half of data set-4 for training and using a quarter for training. Figure 4.8(a) shows the results for pool fraction. With a quarter of the data set for training, the median ε_a of serial and robust approach increase 0.02 and 0.01 respectively, which are the 32% and 21% rise comparing to the median ε_a using half of the data set-4 for training, however, the median ε_a of integrated approach only increases 0.003, which is only 6% raise. In Figure 4.8(b), performance for mass in pools is shown. In both cases, the use of the Huber norm has a significant impact. When training data is plentiful, as shown by the dotted lines in Figure 4.8(b), the integrated approach has little gain (e.g. 0.5 kg difference at 90th percentile of ε_a), but when the training data are scarce, as shown by the solid lines, the gain at 90th percentile of ε_a is almost 3 kg. The mass of ganglia estimation with 50% training data is shown in Figure 4.8(c) by dotted lines. Here we see that the performance of robust approach is almost the same as the integrated approach, but shown by solid lines, where only 25% of the data are used for training, the superior performance of integrated approach is obvious.

In Figure 4.9, we also show the 10th to 90th percentile of relative error for three approaches using half of data set-4 and a quarter for training. Figure 4.9(a) shows the results for pool fraction. With a quarter for training, the median ε_r of serial and robust approach increase 1.88% and 2.35% respectively, however, the median ε_r of the integrated approach only increases 0.34%. In Figure 4.8(b), performance for mass in pools is shown. When

training data is plentiful, as shown by the dotted lines in Figure 4.8(b), the integrated approach has little gain (e.g. 21.69% difference at 90th percentile of ε_r), but when the training data are scarce, as shown by solid lines, the gain at 90th percentile of ε_r is almost 60.40%. The mass of ganglia estimation with 50% training data is shown in Figure 4.8(c) by the dotted lines. Here we see that the performance of integrated approach using 25% of data set for training, shown by red solid lines, is almost the same as the integrated approach using half of data set for training shown by red dotted lines, this demonstrates the superior performance of integrated approach under scarce data set.

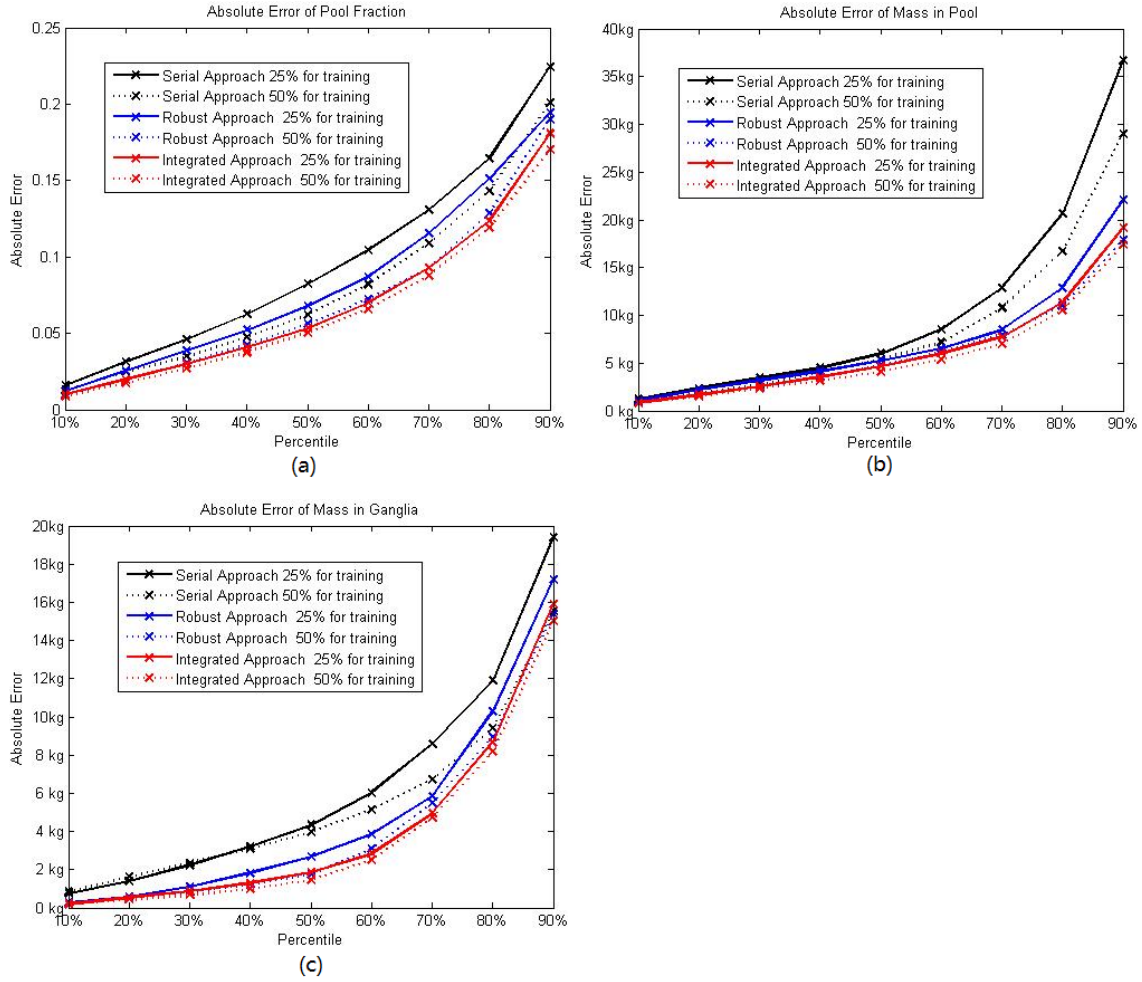

Figure 4.8: The 10th to 90th percentile of absolute error. The results shown by dotted lines are obtained by using half of data set 4 for training and that shown by solid lines are using a quarter of data set 4 for training. Sub-figure (a) shows the statistical result for pool fraction estimation, and (b) for mass in pools and (c) for mass in ganglia respectively. The maximum values of mass in pools and ganglia are listed in Table 4.8

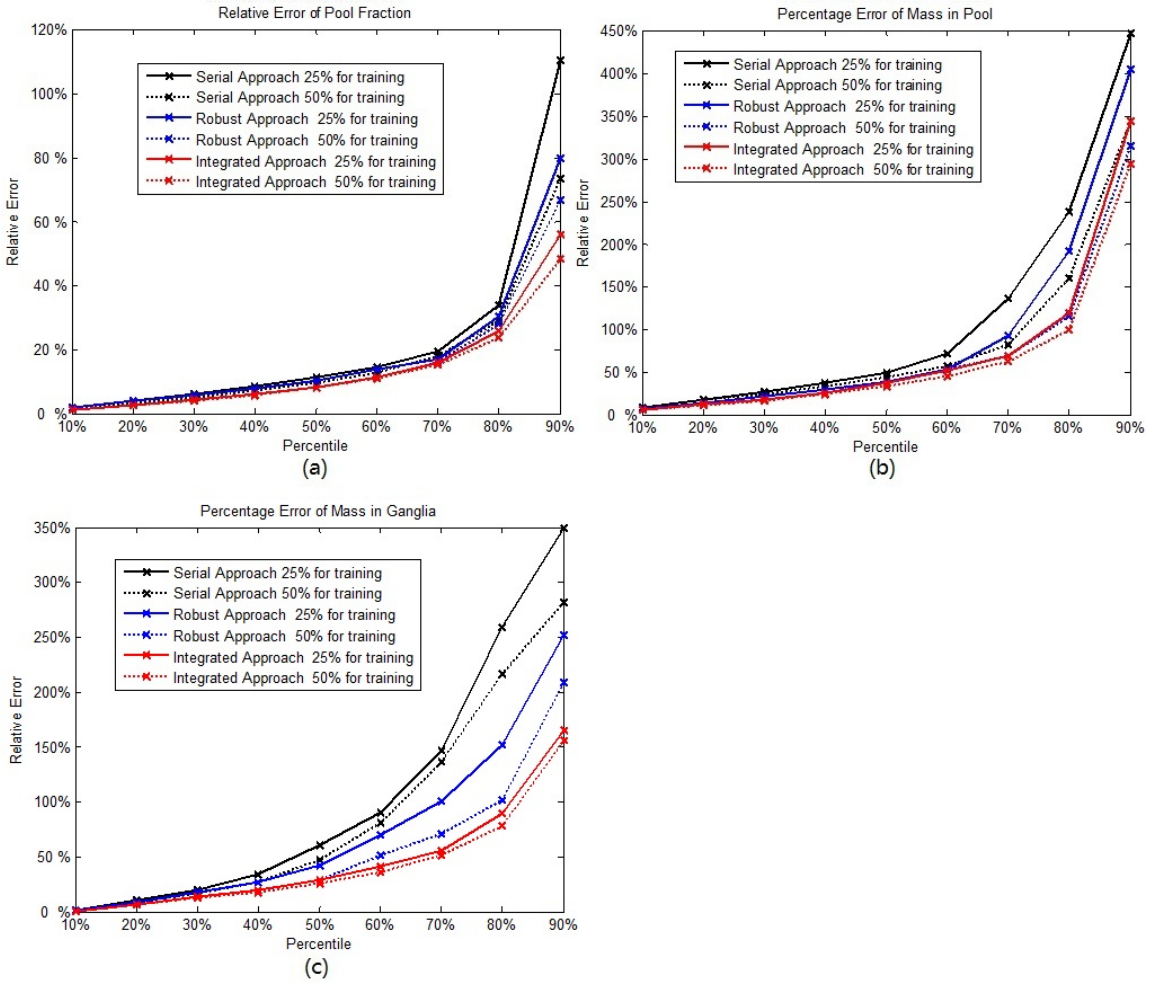

Figure 4.9: The 10th to 90th percentile of relative error. The results shown by dotted lines are obtained by using half of data set 4 for training and that shown by solid lines are using a quarter of data set 4 for training. Sub-figure (a) shows the statistical result for pool fraction estimation, and (b) for mass in pools and (c) for mass in ganglia respectively.

Chapter 5

Sparse Concentration Data

In this chapter, we address an application of our manifold regression approaches for estimating the metrics describing the source zone based upon sparse concentration data. In Chapter 4, we propose three manifold regression approaches for characterizing the source zone based upon observations at a single instant in time of down-gradient DNAPL concentration data collected across a transect oriented perpendicular to the flow. In that scenario, we assumed that the transect data are sampled densely forming an "image" that could be used both for learning and testing the regression functions. In practice, this dense data assumption is not a problem for training, which will be based on high quality simulation results. For testing purpose, however, it is problematic as field data are typically sampled quite sparsely in space [73]. The goal of this chapter then is the development of a metric estimation scheme that allows for "full data" training but "sparse data" testing.

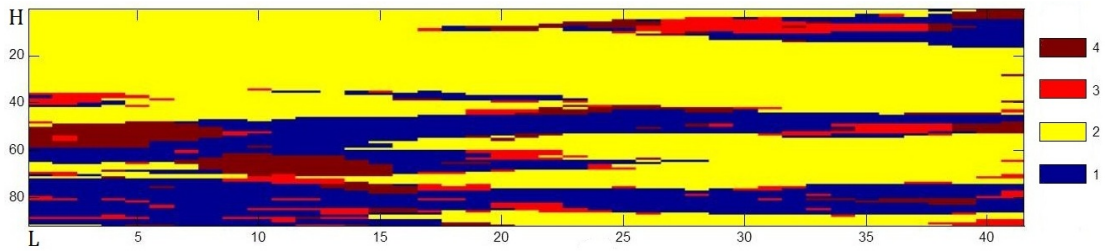

Figure 5.1: The hydraulic conductivity of 2D model, it is comprised of four components which is indicated by number from 1 to 4 with increasing hydraulic conductivity.

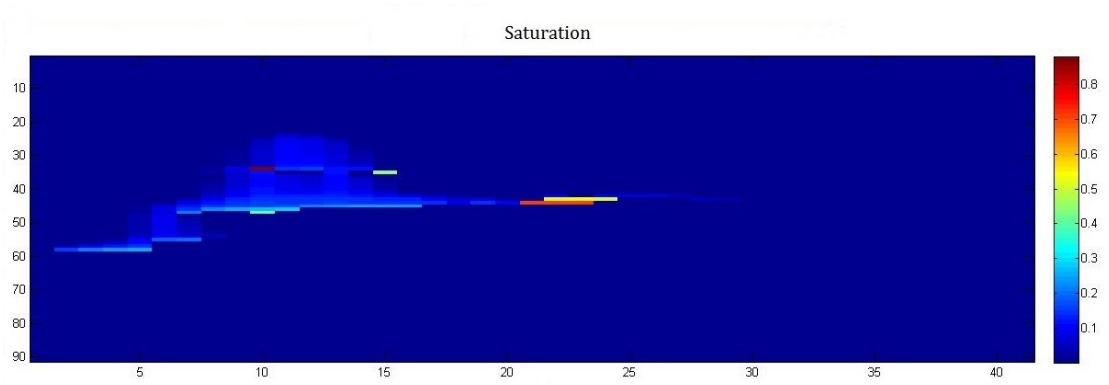

Figure 5.2: The saturation is modeled as being comprised of two parts: "pool" for which the saturation exceeds 0.15 and "ganglia" for which the saturation is lower than 0.15, the color bar indicates the saturation value.

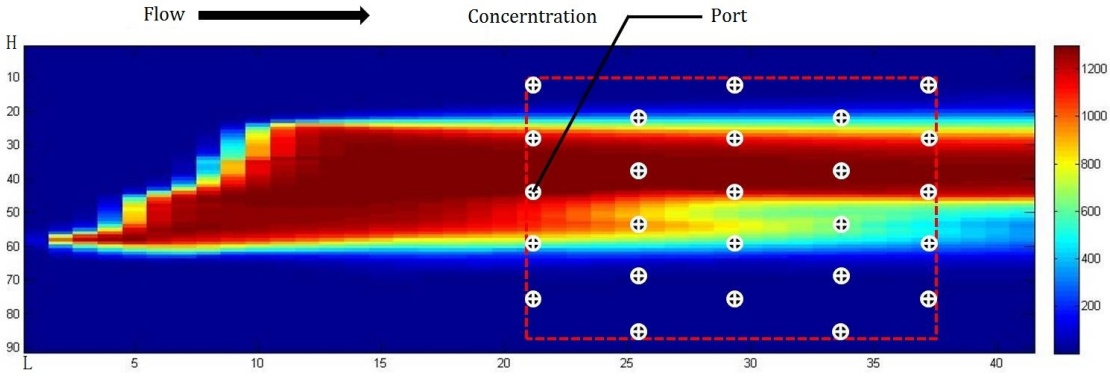

Figure 5.3: The flow through the source zone gives the concentration image, the color bar shows the concentration value (mg/L). The size of 2D model is $H \times L = 48$ cm $\times$ 1 m.

5.1 TP/MC Model

Motivated by the experimental work in our group [73], here we consider a 2D scenario illustrated in Figure 5.1 to Figure 5.3. In Figure 5.1, the hydraulic conductivity field includes four components which represent four kinds of lithofacies with different conductivity. The property of these lithofacies are introduced in Table 2.3. They are represented by number from 1 to 4 with increasing conductivity. The hydraulic conductivity field is generated by Probability Transition/Markov Chain (TP/MC) model [16] which is discussed in detail in Section 2.6.2. In Figure 5.2, the source zone is located on the left side of the region and "groundwater" flow proceeds from left to right. After the DNAPL is released, they reside above the low conductivity layer governed by the multi-phase flow and transport model discussed in Section 2.1. Given the data acquisition scenario of interest here, one natural adaptation of the approaches proposed in Chapter 4 would employ concentration data from the 1300 pixels in the red box of Figure 5.3 for both training and testing, but the sparse samples of test data are too scarce (25 out of 1300 pixels) to reconstruct the full concentration data. Therefore, our goal here is to develop a machine learning method for metric estimation where training proceeds as before but testing is based on samples from the 25 ports[1] shown in Figure 5.3.

5.2 Manifold Regression Framework for Sparse Data

In this chapter, our work focuses on extending the ideas proposed in Chapter 4 to construct the regression functions for estimating the three metrics M_p, M_g and f_p from the sparse

[1]The term "port" reflects nature of the experimental setup in [73] where fluid is extracted from the test cell at these locations.

sampled concentration data. This process is comprised of a number of component steps that are illustrated in Figure 5.4. Several modules are the same as our serial approach shown in Figure 4.1. As the number of our training data, N, is about 3500 and the full concentration image for training include 1300 pixels, we must first reduce the dimensionality of the data. Here we follow the same two-steps method to reduce the dimensionality as the serial approach in Chapter 4: feature extraction from the concentration images followed by a low dimensional, manifold-based representation of this feature set. In Chapter 3, we proposed a set of geometric features $\mathbf{x}$ extracted from the concentration using ideas from morphological image processing operations [28], the structure of which are motivated by extracting the shape and value information of concentration image. The SGS data sets are sampled on the transect which perpendicular to the flow, however, the data used in this chapter are sampled along the flow direction. But through our observation, the shape and value information of concentration are still predictive to the estimation of metrics. Thus we still use geometric feature extraction method to convert the concentration image in red box of Figure 5.3 to the feature vector. In order to transform the feature vectors into a space such that the distance between feature vectors is reflective of the distance between the corresponding metrics, we still employ Laplacian Eigenmaps (LE) [9] to obtain a low dimensional manifold coordinate vector $\mathbf{r}$ for training a regression function under a Bayesian approach.

In Chapter 4 we considered the case of a manifold constructed in a supervised manner using both the full geometric feature vector $\mathbf{x}$ as well as the known metrics $\mathbf{t}$. Spectral Regression is then employed to embed test data comprised only of the feature vector (no metrics) into the manifold. One way to process the sparse test data is full concentration

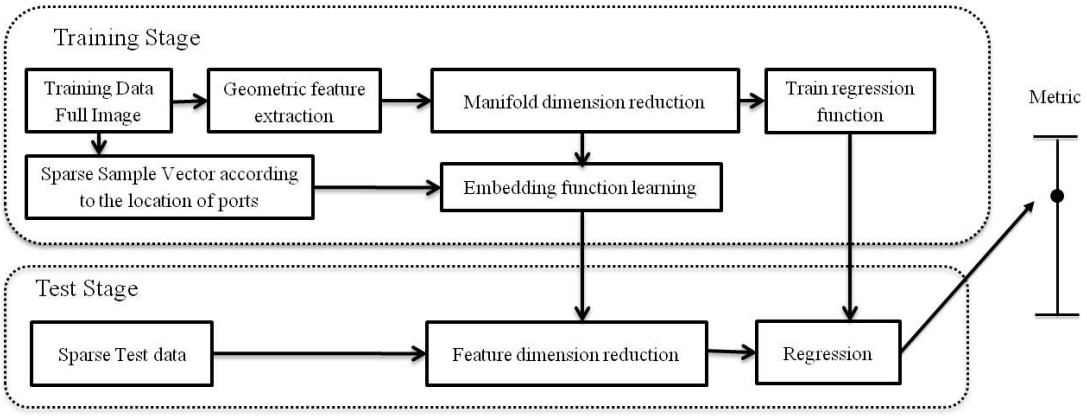

Figure 5.4: The framework of our regression-based machine learning approach using sparse test data. The geometric feature extraction is used only by the training stage, in the test stage the concentration is sparsely sampled according to the position of ports.

image interpolation based on the sparse samples and then extract geometric features from full image, however, the test data in our problem are too sparse (i.e., 25 out of 1300) to reconstruct the full image precisely. But the SR can bypass this issue. In Section 4.2, SR learns an embedding function $\mathbf{r} = f(\mathbf{x}) = \mathbf{A}^T\mathbf{k}(\mathbf{x})$, specified by the parameter matrix $\mathbf{A} \in \Re^{N\times m}$ and kernel function vector $\mathbf{k}(\mathbf{x})$ to map geometric feature vector into an already-built manifold. In this chapter, we build the manifold in the same manner, but now learn the embedding function from the raw 25-dimensional sparse sample vector $\mathbf{x}^{(s)}$ sampled from training image to the manifold coordinates $\mathbf{r}$ using SR,

$$\mathbf{A}^* = \underset{\mathbf{A}}{\operatorname{argmin}} = \sum_{i=1}^{N} \|\mathbf{A}^T\mathbf{k}(\mathbf{x}_i^{(s)}) - \mathbf{r}_i\|_2^2 + \gamma\|\mathbf{A}\|_F^2 \tag{5.1}$$

Surprisingly, despite the rather substantial difference between the quantities used to construct the manifold and those available for learning embedding function, there is relatively little degradation in our ability to determine the metrics which is illustrated in Section 5.3.2.

We also use the two new proposed manifold regression approaches in Chapter 4 to

process the sparse concentration data. The robust approach changes the $L2$ norm in (5.1) to the Huber norm. The integrated approach jointly determines the spectral regression and Bayesian regression functions needed to embed test data into the manifold and estimate the metrics respectively. Mathematically, we only have to change the $\mathbf{k}(\mathbf{x}_i)$ in (4.7) to $\mathbf{k}(\mathbf{x}_i^{(s)})$ where $\mathbf{x}_i^{(s)}$ is the sparse concentration datum vector.

5.3 Experiments

5.3.1 Data Set

The random hydraulic conductivity field of 2D model is generated by Transitional Probability Markov Chain (TP/MC) using statistical parameters from Maji's work [49] which is discussed in Section 5.1. M-VALOR [73] was used to simulate the infiltration and entrapment of contaminant in 2D aquifer cells. Dissolution and TCE solute transport were modeled with a modified form of MT3DMS [80]. The training data are comprised of the full concentration images within the red bounding box shown in Figure 5.3 for the various realizations. The test data are 25-dimensional sparse sample vectors taken from the indicated ports.

5.3.2 Experimental Results

We evaluate the performances of three approaches to metrics estimation. Under the *serial approach*, each module in Figure 5.4 is constructed independently using the ideas discussed in Section 5.2. We compare this serial approach with the *robust approach* and the fully *integrated approach* to verify that integrated approach can improve the accuracy of the

metrics estimation. We randomly select half of data set ($N = 3500$) for training and the rest of data set for test, repeating this procedure 5 times. The hyper-parameter selection methods in Section 4.6.2 are used to determine the γ_i's in (5.1) and the σ_1, σ_2 and σ_{SR} in the LE and SR kernel functions. Additionally, to compare the performance degeneration using sparse test data, we also apply our approaches based on full concentration test data. Table 5.1 presents the range of metrics and the statistical results of the serial, robust and integrated approaches, where $\varepsilon_a = |t - \hat{t}|$ is the absolute value of the difference between the estimation and true metric and $\varepsilon_r = \varepsilon_a/t \times 100\%$ is the relative error. EP85 is empirical percentage of true metrics falling into the 85% confidence intervals.

	The Range of Metric	Algorithm	Test Data	median ε_a	median ε_r	EP85
f_p	$0 \sim 1$	Serial Approach	Full	0.0451	4.63%	93.55%
			Sparse	0.0459	4.74%	92.69%
		Robust Approach	Full	0.0339	3.44%	93.78%
			Sparse	0.0404	4.19%	92.74%
		Integrated Approach	Full	0.0228	2.41%	91.44%
			Sparse	0.0274	2.84%	91.84%
M_p	$0 \sim 57.3$ (g)	Serial Approach	Full	5.15	38.1%	89.08%
			Sparse	5.27	39.0%	81.77%
		Robust Approach	Full	3.35	28.7%	82.28%
			Sparse	4.71	35.6%	83.34%
		Integrated Approach	Full	3.14	24.4%	83.79%
			Sparse	3.99	30.3%	85.13%
M_g	$0 \sim 23.8$ (g)	Serial Approach	Full	0.649	62.0%	86.84%
			Sparse	0.750	67.2%	84.72%
		Robust Approach	Full	0.289	41.9%	90.83%
			Sparse	0.457	55.8%	90.71%
		Integrated Approach	Full	0.232	33.6%	91.10%
			Sparse	0.280	40.8%	92.43%

Table 5.1: The statistical result using half of the data set for training, two types of test data are applied for evaluating the performance of regression functions. One is full image which is shown within the red rectangle in Figure 5.3, the other is sparse signal sampled from ports in Figure 5.3.

From the results, the median error using sparse test data is only a little larger than that using full test data. The median error of f_p estimation using integrated approach based on sparse test data is 0.0274, the median error using integrated approach based on full test data is 0.0228, thus the degeneration is only 0.0046. The degeneration of M_p estimation is from 3.14 g to 3.99 g and the degeneration of M_g estimation is from 0.232 g to 0.280 g. This validates the hypothesis that metrics of interest can be recovered very accurately from sparse sampled data. The median error of integrated approach is almost half of that for the serial approach. The median error of f_p estimation drops from 0.0459 to 0.0274 by using integrated approach, the median error of M_p drops from 5.27 g to 3.99 g and the median error of M_g changes from 0.750 g to 0.280 g. This demonstrates the efficacy and superior performance of our integrated manifold regression approach which can simultaneously determine the embedding function and three regression functions.

Chapter 6

Conclusion and Future Work

In this thesis, we propose two machine-learning frameworks to address the problem of contaminant characterization in source zone. Accidentally released hazardous Dense Non-Aqueous Phase Liquid poses the persistent danger to drinking water aquifers. A critical component in the planning of a remediation approach and the monitoring of the cleanup effort is characterizing the source zone. The problem of source zone characterization is complicated by the fact that the distribution of contaminant is determined to a large extent by the spatial variability in hydraulic conductivity, which is typically modeled as a random process whose statistics may be known for a given site but whose specific spatial distribution is certainly not known. Hydrological scientists found that a single metric pool fraction f_p of the subsurface is necessary for remediation planning, we also found two other metrics mass of DNAPL in pools M_p and mass of DNAPL in ganglia M_g are also interesting for contaminant characterization, these metrics follow the constraint $f_p = M_p/(M_p + M_g)$. Therefore, in this work we use machine learning algorithms to estimate these three metrics based on the concentration data sampled at down-gradient. Given the availability of statistical models of

the conductivity along with numerical models for both DNAPL entrapment and subsequent flow and transport, hydrological scientists can simulate a large number of conductivity fields, spill scenarios, and observations of down-gradient concentration. These data are used for training the classifiers and regression functions for estimating the metrics.

First we propose a set of geometric features extracted from concentration images. Since the observations of contaminant concentrations are located along a transect down-gradient from the source zone, we seek the features that capture the size and number of blobs in the concentration data which are more predictive for the estimation of metrics than the raw images. This feature extraction is a kind of image processing methods, not involving any training procedure and is appropriate and useful for both the classification and regression methods.

Second, we have developed a classification approach to determining the parameters of interest. Rather than point estimation for each metric, the method gives a metric interval for each concentration datum. In the training procedure, after the feature extraction, we propose an iterative Linear Discriminant Analysis and Spectral Clustering algorithm (LDA-SC algorithm) which are employed to reduce the dimension of feature vectors and cluster them in a reduced feature space. Since the metrics are continuously valued, after metric discretization we divide the metric into several non-overlapping bins, each of which represents a class. The k-nearest-neighbor method is employed to classify the test datum. Our iterative LDA-SC algorithm is compared against the classic Principle Component Analysis and K-means method, the performance of our new algorithm demonstrates the superior classification ability of our approach.

Furthermore, we propose a manifold regression framework to solve the challenge problem

which is giving point estimations to the three metrics characterizing the structure of a subsurface contaminant source zone given the observations of down-gradient concentration. The training process of the regression framework includes a number of steps. After the geometric feature extraction, we employ Laplacian Eigenmaps to reduce the dimension of feature vectors and in the manifold space the data with similar metrics vector will be located near-by. In this space, we use Bayesian regression method to train the regression function which can give both the estimation and the confidence interval. Since LE needs both the feature and metrics vector to embed the training data, we apply Spectral Regression (SR) method for learning the embedding function to embed the test data without the metrics information into the same manifold as training data. In the test process of regression framework, the test datum is first embedded into the manifold space and then the regression functions provide the point estimations of the metrics. Due to the existence of outliers, we proposed a robust variant of SR to find a robust embedding function which is less sensitive to the outliers. In the integrated approach, in addition to the robustness we use the mathematical relationship among three metrics as a constraint to improve performance. The experiments using Sequential Gaussian Simulation (SGS) data sets validate the performance of the approaches we proposed in this work showing that the metrics of interest can be recovered very accurately even from limited data.

We also apply our serial approach, robust approach and integrated approach to estimate the metrics f_p, M_p and M_g given sparse observations of concentration. A different hydraulic conductivity model, Transition Probability Markov Chain (TP/MC) model, gives rise to piecewise constant hydraulic conductivity fields which should result in qualitative structure of the DNAPL distribution that differs markedly from that obtained using the SGS model.

The concentration data for test are sparsely sampled rather than the full images used for SGS data. The experiments using TP/MC simulation data sets validate the performance of the approaches we proposed in this work showing that the metrics of interest can be recovered very accurately even from sparse sampled data.

Given these results, there are a variety of issues needed to be explored in the future. For the classification framework, in the LDA-SC algorithm, we first reduce the dimension of feature space and cluster in the reduced feature space, then discretize the metric. In the future, we may incorporate the metric discretization into clustering, thus in the reduced feature space, we can simultaneously find the non-overlapping clusters and metric intervals. This may improve the performance of our classifier. Moreover, since the metric interval corresponding to each cluster is overlap, the linear dimension reduction method such as LDA may be not an appropriate choice. In the future, kernel LDA [61] can be used for dimension reduction. Another future work for classification framework can be the proof of convergence of our LDA-SC algorithm.

For manifold regression framework, from the machine learning perspective, the first issue we need to explore in the future is to find a method to select hyper-parameters more efficiently which is great interesting because grid search grows exponentially and become exhausted when the number of hyper-parameters increases. Also, we can separate the data set into three categories, one for training, one for hyper-parameter selection and one for testing. Second, the objective function of integrated approach is not convex, thus the solution is not guaranteed to be global optimal, a convex approximation of integrated approach is necessary for future research. Third, mathematical morphology provides a far larger range of features than the two considered in this thesis that may be of use in estimating the

metrics. It would certainly be of interest to explore other options. From the application perspective, first, the SGS concentration data sets used in this work are sampled across the transect at the end of the source zone, however, in the field only far data can be acquired. It is an open question that where the concentration data should be sampled. Using these far sampled data, whether the metrics can be estimated accurately is a challenging question for us in the future. Related to this issue is the need to validate the ideas in this work using real data either from lab-scale experiments or eventually from field sites. Second, application of the machine learning approaches to other types of hydraulic conductivity distributions is also of interest. Third, the only limitation of our SGS data set-4 is that the total volume of DNAPL released in the source zone is fixed for all the realizations. In the future, the data set with different initial volume of contaminant is used for testing the performance of machine-learning approaches.

The focus of this thesis has been a small data problem involving a few thousand training samples. It is certainly possible that the ideas could be applied to big data problem commonly encountered in the geoscience and remote sensing communities. Doing so however may require some consideration of the computational burden associated with the training phase. Specifically, as we indicate in the Appendix A the gradient of L with respect to $\mathbf{A}$ and weight vector $\mathbf{w}_k$ require the multiplication of two kernel matrices: one whose dimension is equal to the number of outlier data as defined in Section 4.4 and a second whose dimension is equal to the number of inlier data. Typically, the size of the second would be much larger than the first. Therefore, the limiting calculation required by the gradient decent approach discussed in Section 4.5 is an $\mathcal{O}(N_i^3)$ matrix-matrix multiplication where N_i is the number of inlier training data. Such an operation may be prohibitive for on-line

training. Even in an off-line setting there may be interesting work to be done exploring approximate methods for this calculation or developing alternative, more efficient techniques for solving the optimization problem.

Appendix A

Cyclic Decent Algorithm

Take the Huber norm formula into the unconstrained optimization problem (4.9), we have,

$$\begin{aligned}\min_{\mathbf{A},\mathbf{W}} \quad & L(\mathbf{A},\mathbf{W}) = \frac{1}{N}\left(\sum_{i\in\text{normal}} \frac{\|\mathbf{A}^T\mathbf{k}(\mathbf{x}_i)-\mathbf{r}_i\|_2^2}{2\varepsilon_e} + \sum_{i\in\text{outliers}} \frac{1}{e_i}\|\mathbf{A}^T\mathbf{k}(\mathbf{x}_i)-\mathbf{r}_i\|_2^2 - \frac{\varepsilon_e}{2}\right) \\ & + \gamma_2\frac{1}{N}\left(\sum_{i\in\text{normal}} \frac{\|\mathbf{W}^T\mathbf{A}^T\mathbf{k}(\mathbf{x}_i)-\mathbf{t}_i\|_2^2}{2\varepsilon_h} + \sum_{i\in\text{outliers}} \frac{1}{h_i}\|\mathbf{W}^T\mathbf{A}^T\mathbf{k}(\mathbf{x}_i)-\mathbf{t}_i\|_2^2 - \frac{\varepsilon_h}{2}\right) \\ & + \gamma_1\|\mathbf{A}\|_F^2 + \gamma_3\|\mathbf{W}\|_2^2 \\ & + \gamma_P\sum_{i=1}^{N}(\mathbf{w}_1\mathbf{A}^T\mathbf{k}(\mathbf{x}_i) + \bar{f}_p - \frac{\mathbf{w}_2\mathbf{A}^T\mathbf{k}(\mathbf{x}_i)+\bar{M}_p}{\mathbf{w}_2\mathbf{A}^T\mathbf{k}(\mathbf{x}_i)+\bar{M}_p+\mathbf{w}_3\mathbf{A}^T\mathbf{k}(\mathbf{x}_i)+\bar{M}_g})^2.\end{aligned} \tag{A.1}$$

where $e_i = \|\mathbf{A}^T\mathbf{k}(\mathbf{x}_i)-\mathbf{r}_i\|_2$ and $h_i = \|\mathbf{W}^T\mathbf{A}^T\mathbf{k}(\mathbf{x}_i)-\mathbf{t}_i\|_2$. The ε_e and ε_h are the thresholds in the Huber norm which are defined in Section 4.5. For convenience, we write (A.1) as

$$\min_{\mathbf{A},\mathbf{W}} \quad L(\mathbf{A},\mathbf{W}) = J(\mathbf{A},\mathbf{W}) + \gamma_P P(\mathbf{A},\mathbf{W}) \tag{A.2}$$

The gradient of J with respect to $\mathbf{A}$ is

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{A}} = & \frac{2}{N}\left(\frac{1}{\varepsilon_e}\mathbf{A}^T\mathbf{K}_n\mathbf{K}_n^T - \frac{1}{\varepsilon_e}\mathbf{R}_n\mathbf{K}_n^T + \mathbf{A}^T\mathbf{K}_o\mathbf{E}\mathbf{K}_o^T - \mathbf{R}_o\mathbf{E}\mathbf{K}_o^T\right) + 2\gamma_1\mathbf{A}^T \\ & + \frac{2\gamma_2}{N}\left(\frac{1}{\varepsilon_h}\mathbf{W}\mathbf{W}^T\mathbf{A}^T\mathbf{K}_n\mathbf{K}_n^T - \frac{1}{\varepsilon_h}\mathbf{W}\mathbf{T}_n\mathbf{K}_n^T + \mathbf{W}\mathbf{W}^T\mathbf{A}^T\mathbf{K}_o\mathbf{H}\mathbf{K}_o^T - \mathbf{W}\mathbf{T}_o\mathbf{H}\mathbf{K}_o^T\right)\end{aligned} \tag{A.3}$$

where $\mathbf{K}_n$ is the kernel matrix of inliers and $\mathbf{K}_o$ is the kernel matrix of outliers. The matrices $\mathbf{R}_n$ and $\mathbf{R}_o$ are the manifold coordinate matrices of inliers and outliers respectively. The matrices $\mathbf{T}_n$ and $\mathbf{T}_o$ are the metrics vectors of inliers and outliers respectively. The matrix $\mathbf{E}$ is the diagonal matrix of $\frac{1}{e_i}$ and $\mathbf{H}$ is the diagonal matrix of $\frac{1}{h_i}$.

The gradient of J respect to $\mathbf{w}_k$ where $k = 1, 2, 3$ is,

$$\frac{\partial J}{\partial \mathbf{w}_k} = \frac{2\gamma_2}{N}\left(\frac{1}{\varepsilon_h}\mathbf{A}^T\mathbf{K}_n\mathbf{K}_n^T\mathbf{A}\mathbf{w}_k - \frac{1}{\varepsilon_h}\mathbf{A}^T\mathbf{K}_n\mathbf{t}_n^{(k)} + \mathbf{A}^T\mathbf{K}_o\mathbf{H}\mathbf{K}_o^T\mathbf{A}\mathbf{w}_k - \mathbf{A}^T\mathbf{K}_o\mathbf{H}\mathbf{t}_o^{(k)}\right) + 2\gamma_3\mathbf{w}_k \tag{A.4}$$

where $\mathbf{t}^{(k)}$ is a metric vector, (e.g. $\mathbf{t}^{(1)}$ is comprised of f_p for each training data).

We rewrite the regularization term $P(\mathbf{A}, \mathbf{W}) = \sum_{i=1}^{N}(\hat{f}_p - \frac{\hat{M}_p}{\hat{M}_p + \hat{M}_g})^2$ as $P(\mathbf{A}, \mathbf{W}) = \sum_{i=1}^{N}((1 - \hat{f}_p)\hat{M}_p - \hat{f}_p\hat{M}_g)^2$. We take the gradient of $P(\mathbf{A}, \mathbf{W})$ with respect to $\mathbf{A}$ is,

$$\frac{\partial P}{\partial \mathbf{A}} = 2\sum_{i=1}^{N}\left([(1-\hat{f}_p)\hat{M}_p - \hat{f}_p\hat{M}_g]\left(-\mathbf{w}_1\mathbf{k}(\mathbf{x}_i)\hat{M}_p + (1-\hat{f}_p)\mathbf{w}_2\mathbf{k}(\mathbf{x}_i) - \hat{M}_g\mathbf{w}_1\mathbf{k}(\mathbf{x}_i) - \hat{f}_p\mathbf{w}_3\mathbf{k}(\mathbf{x}_i)\right)\right) \tag{A.5}$$

The gradient of $P(\mathbf{A}, \mathbf{W})$ with respect to $\mathbf{w}_1$ is,

$$\frac{\partial P}{\partial \mathbf{w}_1} = 2\sum_{i=1}^{N}\left([(1 - \hat{f}_p)\hat{M}_p - \hat{f}_p\hat{M}_g](-\mathbf{A}^T\mathbf{k}(\mathbf{x}_i)\hat{M}_p - \mathbf{A}^T\mathbf{k}(\mathbf{x}_i)\hat{M}_g)\right) \tag{A.6}$$

the gradient of $P(\mathbf{A}, \mathbf{W})$ with respect to $\mathbf{w}_2$ is,

$$\frac{\partial P}{\partial \mathbf{w}_2} = 2\sum_{i=1}^{N}\left([(1-\hat{f}_p)\hat{M}_p - \hat{f}_p\hat{M}_g](1-\hat{f}_p)\mathbf{A}^T\mathbf{k}(\mathbf{x}_i)\right) \tag{A.7}$$

and the gradient of $P(\mathbf{A}, \mathbf{W})$ with respect to $\mathbf{w}_3$ is,

$$\frac{\partial P}{\partial \mathbf{w}_3} = 2\sum_{i=1}^{N}\left([(1-\hat{f}_p)\hat{M}_p - \hat{f}_p\hat{M}_g](-\hat{f}_p\mathbf{A}^T\mathbf{k}(\mathbf{x}_i))\right) \tag{A.8}$$

Bibliography

[1] Linda M Abriola. Modeling multiphase migration of organic chemicals in groundwater systems–a review and assessment. *Environmental Health Perspectives*, 83:117, 1989.

[2] Linda M Abriola, Chad D Drummond, Ernest J Hahn, Kim F Hayes, Tohren CG Kibbey, Lawrence D Lemke, Kurt D Pennell, Erik A Petrovskis, C Andrew Ramsburg, and Klaus M Rathfelder. Pilot-scale demonstration of surfactant-enhanced pce solubilization at the bachman road site. 1. site characterization and test design. *Environmental science & technology*, 39(6):1778–1790, 2005.

[3] Alireza Aghasi. *Parametric Shape Based Methods for Inverse Problems*. PhD thesis, Tufts University, 2012.

[4] Alireza Aghasi, Misha Kilmer, and Eric L Miller. Parametric level set methods for inverse problems. *SIAM Journal on Imaging Sciences*, 4(2):618–650, 2011.

[5] Leona S Aiken, Stephen G West, and Steven C Pitts. Multiple linear regression. *Handbook of psychology*, 2003.

[6] John Robert Anderson, Ryszard Spencer Michalski, Ryszard Stanisław Michalski, Thomas Michael Mitchell, et al. *Machine learning: An artificial intelligence approach*,

volume 2. Morgan Kaufmann, 1986.

[7] P Bayer, P Huggenberger, Philippe Renard, and A Comunian. Three-dimensional high resolution fluvio-glacial aquifer analog: Part 1: Field study. *Journal of Hydrology*, 405(1):1–9, 2011.

[8] Jacob Bear. *Dynamics of fluids in porous media*. Courier Dover Publications, 2013.

[9] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[10] Yoshua Bengio, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in neural information processing systems*, 16:177–184, 2004.

[11] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[12] Deng Cai, Xiaofei He, and Jiawei Han. Spectral regression: A unified approach for sparse subspace learning. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 73–82. IEEE, 2007.

[13] Gustavo Camps-Valls, T Bandos Marsheva, and Dengyong Zhou. Semi-supervised graph-based hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(10):3044–3054, 2007.

[14] Gustavo Camps-Valls and Lorenzo Bruzzone. Kernel-based methods for hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(6):1351–1362, 2005.

[15] Gustavo Camps-Valls, Luis Gomez-Chova, Jordi Muñoz-Marí, Joan Vila-Francés, and Javier Calpe-Maravilla. Composite kernels for hyperspectral image classification. *Geoscience and Remote Sensing Letters, IEEE*, 3(1):93–97, 2006.

[16] Steven F Carle and Graham E Fogg. Modeling spatial variability with one and multidimensional continuous-lag markov chains. *Mathematical Geology*, 29(7):891–918, 1997.

[17] Jianhui Chen, Jieping Ye, and Qi Li. Integrating global and local structures: A least squares framework for dimensionality reduction. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[18] Yangchi Chen, Melba M Crawford, and Joydeep Ghosh. Applying nonlinear manifold learning to hyperspectral data for land cover classification. In *IGARSS*, volume 5, pages 24–29, 2005.

[19] John A Christ, Lawrence D Lemke, and Linda M Abriola. The influence of dimensionality on simulations of mass recovery from nonuniform dense non-aqueous phase liquid (dnapl) source zones. *Advances in Water Resources*, 32(3):401–412, 2009.

[20] John A Christ, C Andrew Ramsburg, Linda M Abriola, Kurt D Pennell, and Frank E Löffler. Coupling aggressive mass removal with microbial reductive dechlorination for remediation of dnapl source zones: a review and assessment. *Environmental Health Perspectives*, 113(4):465, 2005.

[21] John A Christ, C Andrew Ramsburg, Kurt D Pennell, and Linda M Abriola. Estimating mass discharge from dense nonaqueous phase liquid source zones using upscaled mass transfer coefficients: An evaluation using multiphase numerical simulations. *Water Resources Research*, 42(11), 2006.

[22] John A Christ, C Andrew Ramsburg, Kurt D Pennell, and Linda M Abriola. Predicting dnapl mass discharge from pool-dominated source zones. *Journal of contaminant hydrology*, 114(1):18–34, 2010.

[23] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[24] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[25] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 1990.

[26] Zoubin Ghahramani. Unsupervised learning. In *Advanced Lectures on Machine Learning*, pages 72–112. Springer, 2004.

[27] Luis Gómez-Chova, Gustavo Camps-Valls, Jordi Munoz-Mari, and Javier Calpe. Semisupervised image classification with laplacian support vector machines. *Geoscience and Remote Sensing Letters, IEEE*, 5(3):336–340, 2008.

[28] Rafael C Gonzalez, Richard E Woods, and Steven L Eddins. *Digital image processing using MATLAB*, volume 2. Gatesmark Publishing Knoxville, 2009.

[29] Antoine Guitton and William W Symes. Robust inversion of seismic data using the huber norm. *Geophysics*, 68(4):1310–1319, 2003.

[30] Guodong Guo, Yun Fu, Charles R Dyer, and Thomas S Huang. Image-based human age estimation by manifold learning and locally adjusted robust regression. *Image Processing, IEEE Transactions on*, 17(7):1178–1188, 2008.

[31] Joseph C Harsanyi and C-I Chang. Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *Geoscience and Remote Sensing, IEEE Transactions on*, 32(4):779–785, 1994.

[32] Jürgen Heinz, Sybille Kleineidam, Georg Teutsch, and Thomas Aigner. Heterogeneity patterns of quaternary glaciofluvial gravel bodies (sw-germany): application to hydrogeology. *Sedimentary geology*, 158(1):1–23, 2003.

[33] Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1), 2010.

[34] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[35] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[36] Wonkook Kim and Melba M Crawford. Adaptive classification for hyperspectral image data using manifold regularization kernel machines. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(11):4110–4121, 2010.

[37] Boris Kostic, Andreas Becht, and Thomas Aigner. 3-d sedimentary architecture of a quaternary gravel delta (sw-germany): Implications for hydrostratigraphy. *Sedimentary Geology*, 181(3):147–171, 2005.

[38] Mark L Kram, Arturo A Keller, Joseph Rossabi, and Lorne G Everett. Dnapl characterization methods and approaches, part 1: Performance comparisons. *Ground Water Monitoring & Remediation*, 21(4):109–123, 2001.

[39] Oliver Kramer. Unsupervised k-nearest neighbor regression. *arXiv preprint arXiv:1107.3600*, 2011.

[40] B Boser Le Cun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.

[41] Lawrence D Lemke and Linda M Abriola. Modeling dense nonaqueous phase liquid mass removal in nonuniform formations: Linking source-zone architecture and system response. *Geosphere*, 2(2):74–82, 2006.

[42] Lawrence D Lemke, Linda M Abriola, and Pierre Goovaerts. Dense nonaqueous phase liquid (dnapl) source zone characterization: Influence of hydraulic property correlation on predictions of dnapl infiltration and entrapment. *Water Resources Research*, 40(1), 2004.

[43] Ming Li and Baozong Yuan. 2d-lda: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters*, 26(5):527–532, 2005.

[44] Yui Man Lui. Advances in matrix manifolds for computer vision. *Image and Vision Computing*, 30(6):380–388, 2012.

[45] Dalton Lunga and Okan Ersoy. Multidimensional artificial field embedding with spatial sensitivity. 2014.

[46] Dalton Lunga, Saurabh Prasad, M Crawford, and Okan Ersoy. Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning. *Signal Processing Magazine, IEEE*, 31(1):55–66, 2014.

[47] Li Ma, Melba M Crawford, and Jinwen Tian. Local manifold learning-based k-nearest-neighbor for hyperspectral image classification. *IEEE transactions on geoscience and remote sensing*, 48(11):4099–4109, 2010.

[48] R Maji, EA Sudicky, S Panday, and G Teutsch. Transition probability/markov chain analyses of dnapl source zones and plumes. *Ground water*, 44(6):853–863, 2006.

[49] Roudrajit Maji. *Conditional stochastic modelling of DNAPL migration and dissolution in a high-resolution aquifer analog*. ProQuest, 2006.

[50] Morris L Marx and Richard J Larsen. *Introduction to mathematical statistics and its applications*. Pearson/Prentice Hall, 2006.

[51] James W Mercer and Robert M Cohen. A review of immiscible fluids in the subsurface: Properties, models, characterization and remediation. *Journal of Contaminant Hydrology*, 6(2):107–163, 1990.

[52] Grégoire Mercier and Marc Lennon. Support vector machines for hyperspectral image classification with spectral-based kernels. In *Geoscience and Remote Sensing Symposium, 2003. IGARSS'03. Proceedings. 2003 IEEE International*, volume 1, pages 288–290. IEEE, 2003.

[53] Bojan Mohar. *Some applications of Laplace eigenvalues of graphs*. Springer, 1997.

[54] Bojan Mohar and Y Alavi. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2:871–898, 1991.

[55] Thomas Perry, Hongyuan Zha, Patricio Frias, Dadan Zeng, and Mark Braunstein. Supervised laplacian eigenmaps with applications in clinical diagnostics for pediatric cardiology. *arXiv preprint arXiv:1207.7035*, 2012.

[56] Bogdan Raducanu and Fadi Dornaika. A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognition*, 45(6):2432–2444, 2012.

[57] Klaus M Rathfelder, John R Lang, and Linda M Abriola. A numerical model (miser) for the simulation of coupled physical, chemical and biological processes in soil vapor extraction and bioventing systems. *Journal of contaminant hydrology*, 43(3):239–270, 2000.

[58] Douglas Reynolds. Gaussian mixture models. *Encyclopedia of Biometrics*, pages 659–663, 2009.

[59] Sam Roweis. Em algorithms for pca and spca. *Advances in neural information processing systems*, pages 626–632, 1998.

[60] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345, 1999.

[61] Bernhard Scholkopft and Klaus-Robert Mullert. Fisher discriminant analysis with kernels. *Neural networks for signal processing IX*, 1999.

[62] MH Schroth, JD Istok, SJ Ahearn, and JS Selker. Characterization of miller-similar silica sands for laboratory hydrologic studies. *Soil Science Society of America Journal*, 60(5):1331–1339, 1996.

[63] Jared Schuetter and Tao Shi. Multi-sample data spectroscopic clustering of large datasets using nystrom extension. *Journal of Computational and Graphical Statistics*, pages 531–542, 2011.

[64] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[65] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[66] Hans F Stroo, Marvin Unger, C Herb Ward, Michael C Kavanaugh, Catherine Vogel, Andrea Leeson, Jeffrey A Marqusee, and Bradley P Smith. Peer reviewed: Remediating chlorinated solvent source zones. *Environmental Science & Technology*, 37(11):224A–230A, 2003.

[67] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[68] Shang Tan Tu, Jia Yu Chen, Wen Yang, and Hong Sun. Laplacian eigenmaps-based polarimetric dimensionality reduction for sar image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(1):170–179, 2012.

[69] Kuniaki Uto, Takahiro Harano, and Yukio Kosugi. Rice growth state estimation by hyperspectral manifold learning. In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pages 4178–4181. IEEE, 2012.

[70] LJP Van der Maaten, EO Postma, and HJ Van Den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10:1–41, 2009.

[71] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[72] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.

[73] D. Walker, N.L. Cpiro, and K.D. Pennell. Competitive solubilization of trichloroethene and tetrachloroethene from non-aqueous phase liquid mixtures. In *Eighth IAHS International Groundwater Quality Conference*, 2013.

[74] Walter J Weber and Francis A DiGiano. *Process dynamics in environmental systems*. Wiley New York, 1996.

[75] John Shawe-Taylor Christopher KI Williams. The stability of kernel principal components analysis and its relation to the process eigenspectrum. In *Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference*, volume 15, page 383. MIT Press, 2003.

[76] Ruobing Wu, Yizhou Yu, and Wenping Wang. Scale: Supervised and cascaded laplacian eigenmaps for visual object recognition based on nearest neighbors.

[77] Zhaohui Xue, Jun Li, Liang Cheng, and Peijun Du. Spectral–spatial classification of hyperspectral data via morphological component analysis-based image separation.

[78] Hsiuhan Lexie Yang and Melba M Crawford. Exploiting spectral-spatial proximity for classification of hyperspectral data on manifolds. In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pages 4174–4177. IEEE, 2012.

[79] B Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[80] Chunmiao Zheng and P Patrick Wang. Mt3dms: A modular three-dimensional multi-species transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems; documentation and user's guide. Technical report, DTIC Document, 1999.

[81] John S Zogorski, Janet M Carter, Tamara Ivahnenko, Wayne W Lapham, Michael J Moran, Barbara L Rowe, Paul J Squillace, and Patricia L Toccalino. Volatile organic compounds in the nations ground water and drinking-water supply wells. *US Geological Survey Circular*, 1292:101, 2006.