VARIABLE RELEVANCE ASSIGNMENT USING MULTIPLE MACHINE LEARNING METHODS

BY

WEI-WEN FENG

B.S., National Chiao-Tung University, 2002

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2006

Urbana, Illinois

Abstract

With the advance in remote sensing, various machine learning techniques could be applied to study variable relationships. Although prediction models obtained by using machine learning techniques are suitable for predictions, they do not explicitly provide means for determining input-output variable relevance. The relevance information is often of interest to scientists since relationships among variables are unknown.

In this thesis, we investigated the issue of relevance assignment for multiple machine learning models applied to remote sensing variables in the context of terrestrial hydrology. The relevance is defined as the influence of an input variable with respect to predicting the output result. We follow the classical conceptual definition of relevance, and introduce a methodology for assigning relevance using various machine learning methods. The learning methods we use include Regression Tree, Support Vector Machine, and K-Nearest Neighbor. We derive the relevance computation scheme for each learning method, and propose a method for fusing relevance assignment results from multiple learning techniques by averaging and voting mechanism. All methods are evaluated in terms of relevance accuracy estimation with synthetic and measured data. The main contribution of this thesis is a methodology for relevance assignment for multiple learning methods based on local regression, and the fusion methods better robustness.

To My Family.

Acknowledgments

This thesis would not have been possible without the support from many people. Thanks to my thesis advisor, Dr. Peter Bajcsy, who motivates me to work on this project and offers me lots of suggestions and guidances throughout the research. Many thanks to my de jure advisor, Prof. Jiawen Han, who provided valuable feedback in the final stage of the thesis. I would like to thank Prof. Praveen Kumar, Dr. Peter Bajcsy, and David Tcheng, as the principal investigators of the NASA project funding my MS thesis. I also acknowledge NASA for providing the remote sensing data. Special thanks to David Clutter, Rob Kopper, and San-Chul Lee, who have always been patient and generous in helping me learn about machine learning and software design. Thanks to my parents and friends who gave me support when I was stressed. Finally, thanks to my lovely girlfriend, Li-Ying, who endured many dull afternoons with me during the thesis writing, and has always been there for me.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

RA	Relevance Assignment.
RT	Regression Tree.
KNN	K-Nearest Neighbors.
SVM	Support Vector Machine.
PC	Percentage Correct.
DE	Distance Error.
D2K	Data-to-Knowledge system for data mining.
Fpar	Fraction of photosynthetic active radiation.
LAI	Leaf area index.
LST	Land surface temperature.

List of Symbols

R_{ij} Example relevance for input variable v_i of sample e_j.

R_i Model relevance for input variable i.

R_{ij}^{GT} Ground truth example relevance for input variable v_i of sample e_j.

R_i^{GT} Ground truth model relevance for input variable v_i of sample e_j.

R_{ij}^k Example relevance contributed by method k for input variable v_i of sample e_j.

R_i^k Model relevance contributed by method k for input variable v_i.

$Rank\{R_i\}$ The rank of relevance for variable v_i over all input variables.

$\vec{w}$ Hyperplane normal.

$\vec{u_i}$ Basis vector of feature coordinate for variable v_i.

Chapter 1

Introduction

The problem of understanding relationships and dependencies among geographic variables (features) is of high interest to scientists in many data-driven, discovery-type analyses. Various machine learning methods have for data-driven analyses been developed to build prediction models that represent input-output variable relationships. However, prediction models obtained by using machine learning techniques vary in their underlying model representations, and frequently do not provide a clear interpretation of input-output variable relationships. Thus, the goal of data-driven modeling is not only accurate prediction but also interpretation of the input-output relationships.

In this thesis, we address the problem of data-driven model interpretation to establish relevance of input variables with respect to predicted output variables. First, we introduce the previous work in chapter 2, and formulate an interpretation of data-driven models by assigning relevance to input variables in chapter 3. Relevance assignments are derived at the sample (local) or model (global) levels based on co-linearity of input variable basis vectors with the normal of regression hyper-planes formed over model-defined partitions of data samples. Second, we propose algorithms for combining relevance assignments obtained from multiple data-driven models in chapter 4. Finally, we evaluate accuracy of relevance assignment by using (a) three types of synthetic and one set of measured data, (b) three machine learning algorithms, such as Regression Tree (RT), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), and (c) two relevance assignment fusion methods as documented in chapter 5, and summarize our results in chapter 6. The novelty of our work lies in developing a methodology for relevance assignment over a set of machine learning models, proposing relevance fusion methods, and demonstrating the accuracy of multiple relevance assignment methods with multiple synthetic and experimental data sets.

Chapter 2

Previous Work

The first task in establishing input-output relationships is to understand which subset of all considered input variables is important for predicting output variables. This problem is known in the literature as the problem of variable/feature selection. There are multiple approaches to the feature selection problem given multiple variables and their observed values. Based on the given data and the goal of the analysis, we can use either unsupervised or supervised techniques to approach this problem.

For unsupervised technique, one can evaluate the importance of input variables directly by measuring information entropy, data correlation, and etc. The rank of input variables are directly computed based on the metric used for information evaluation. The advantage of unsupervised technique is that it does not require the training of a data-driven model, and is usually computationally faster. However, this evaluation might not reflect the actual input-output relationship since it only uses input variables.

One can also utilize supervised technique by building a data-driven model and evaluating the variable importance based on accuracy of the model. Supervised technique can be blindly used to select a optimal combination of variables that yield the most accuracy model, or it can be used to directly assign the relevance values to the input variables.

In the following sections, we introduce two main directions of achieving feature selection from the previous research literature. One is feature subset selection, which selects the relevant features from the input variables for the learning tasks. The other is dimensionality reduction, which reprojects the original features into a feature space of lower dimension. The learning tasks are thus performed on this reduced feature space. A common technique for achieving this is by using principal component analysis (PCA)[13].

2.1 Feature Subset Selection

Feature subset selection is one of the classical research areas in machine learning [12] [15] [10]. Its goal is to pre-select the most relevant features for learning concepts and to improve accuracy of learning [21]. There are multiple approaches to perform this pre-selection. One can exhaustively search through all feature combinations for the optimal one. To make the problem tractable, the trimmed search can be used to obtain a sub-optimal result. One can also perform the relevance assignments for the input variables, and thresholding the relevance values to obtain a subset of features. In the work of Pudil et al [22], the authors use a sequential search through all possible combinations of feature subsets, and find an optimal feature subset that yields the best result. With a similar attempt, Perner and his coworkers [20] compare the influence of different feature selection methods in the learning performance for decision tree.

However, the exhaustive search over all possible combinations of feature subsets is not always computationally feasible for large number of features, for instance, in the case of hyperspectral band selection. To address the feasibility aspect, Bajcsy and Groves [3] [9] proposed to use a combination of unsupervised and supervised machine learning techniques. A statistical based approach for spectral classification was proposed by S. De Backer et al. [2]. In the work by Kempeneers et al [14], the wavelet based feature transformation is used to achieve high accuracy in band classification.

Another way for selecting the relevant feature subset is to directly assign the relevance values to each input variable, and make the selection on variables with high relevance. The challenge of this relevance assignment is related to the multitude of relevance definitions. For example, relevance of an input variable could be defined by traversing a regression tree model and by summing its weighted occurrence in the tree structure as proposed by

White and Kumar [24]. In the survey by Blum et al [5], the authors give a conceptual definition of a relevant input variable as a variable that will affect the prediction results if it is modified. In our work, while adhering to the conceptual definition of Blum et al [5], we extend the definition of relevance assignment by numerically quantifying relative importance of input variables. Our relevance assignment is related to the work of Heiler et al [11], in which the authors used co-linearity of basis vectors with the normal of a separating hyper-plane obtained from Support Vector Machine (SVM) method as the metric for relevance assignment. We use the co-linearity property in our relevance assignments derived from a set of regression hyper-planes formed over model-defined partitions of data samples.

2.2 Dimensionality Reduction

Dimensionality reduction [1] is to reduce the dimension of original data by reprojecting the original features into a feature space of lower dimension. Instead of selecting a subset of M features from N input features like in feature subset selection, the goal is to find M dimensional representation of N dimensional input features that would preserve the input feature information. Dimensionality reduction techniques commonly used include principal component analysis (PCA), linear discriminant analysis (LDA) and principal feature analysis (PFA).

The principal component analysis (PCA) is widely used for creating relevant features from the linear transformation of original input data to the principal axis. The advantage of PCA is that its choice of transformation satisfies several optimal properties. An important property is that it maximizes the ”spreading”, or variance of the data in the feature space it creates [18]. Therefore it retains the data variation in the original input variables. The mean square error between the lower dimension approximation and the original data is also minimized. One of the concrete applications of PCA for succinct representation is in the problem of face recognition [23]. In this work, PCA is performed on a series input face images to extract the principal features. These features, so called "Eigenface", are then used to represent the face images and for recognition task.

Another dimension reduction technique called linear discriminant analysis (LDA) [19] is also widely used in machine learning community. It is closely related to Fisher's linear discriminant by finding the linear combination of features which best separate two or more classes. The resulting linear combination can therefore be used for feature dimension reduction prior to the training task. An application of LDA for face recognition can be found in the work by Belhumeur et al [4]. In comparison, the main difference between PCA and LDA is that PCA is an unsupervised learning technique, while LDA is a supervised technique that relies on the class labels of input data.

There are pros and cons of applying either feature subset selection or dimensionality reduction for feature selection. Feature subset selection can retain the features in the original space, but the resulting feature sets will consists of redundancy. On the other hand, dimensionality reduction can transform the original feature into the new feature space with minimum redundancy, but can not retain the original features.

Therefore, although both PCA and LDA are useful in dimension reduction by creating relevant features, they have the main disadvantage that the measurements in the original features are transformed into the lower dimension space. These measurements can only be used indirectly in the projected new features. It is sometimes more desirable to use the subset of the original features for analysis, especially when the scientists want to understand the data relationship of certain features. The technique called principal feature analysis(PFA) is proposed based on this motivation [6]. It first performs the dimension reduction and finds the principal components based on the same criteria as PCA. Instead of projecting the original features to the subspace, it exploits the structure of the obtained principal components of a feature set to find a subset of the original feature vector. Therefore, the resulting subset of the features retain the measurement of original forms. However, comparing to PCA, PFA would obtain the feature sets with more redundancy due to the constraints of picking only original features.

2.3 Thesis Work Related to Previous Works

Relevance assignment in this work differs from feature sub-set selection by not assigning binary label to each feature (in or out of the sub-set) but assigning a continuous value that estimates the relevance of an input variable. For example, in Figure 2.1, each input variable v_i has the result r_i from relevance assignment and p_i from feature subset selection. In feature subset selection, each p_i has only binary value 0 or 1 indicating whether it is selected as relevant feature, while in our relevance assignment method, each r_i has continuous value indicating its relevance in resulting prediction. The relevance value is not only continuous over it dynamic range but also might be continuous over time and space depending on a phenomena. Feature subset selection creates categorical description of input relevance while relevance assignment creates a continuous description. With the continuous relevance values over all input variables by using relevance assignment, a threholding can be further performed to obtain the binary feature subset selection.

Our relevance assignment work differs from the previous works of dimensionality reduction as explained in the previous sections. As we can see in Figure 2.1, dimensionality reduction techniques reproject the original variables $v1$, $v2$, and $v3$ into new feature axes $e1$, $e2$, and $e3$. The dimension of the input data is then reduced by selecting only the feature axes e_i with high eigenvalues. Our relevance assignment work directly assigns a continuous relevance values to each input variables v_i, without transforming the original variables

Techniques for direct relevance assignments have been proposed for several machine learning methods. However, most of them target only specific learning methods, and it is difficult to generalize them to other methods. Our relevance assignment methodology based on linear regression extends to multiple machine learning methods that partition input-output examples.

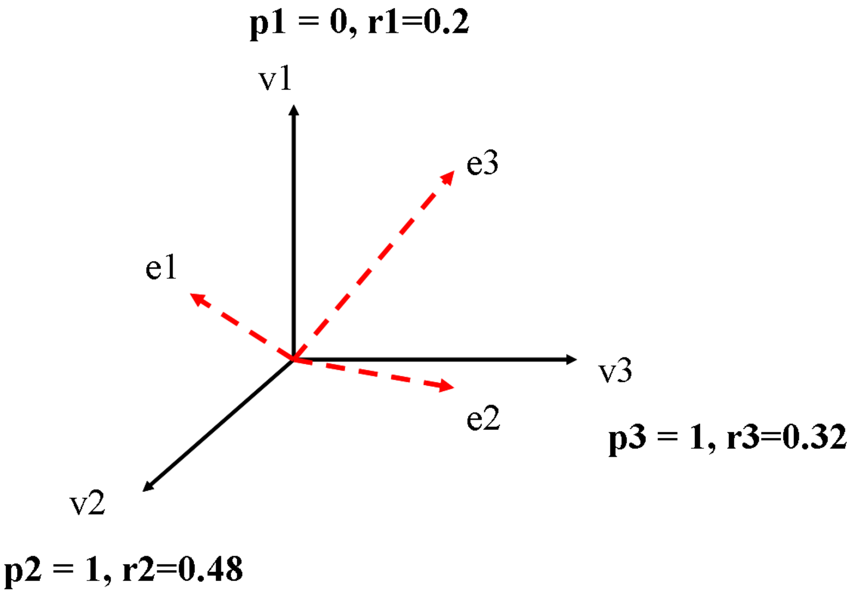


Figure 2.1: Illustration of differences in relevance assignment, feature subset selection, and dimensionality reduction.

Our contribution in this thesis is a relevance assignment methodology derived from previous conceptual definitions [5]. We define the formulation of relevance assignment for three data mining methods, and propose a scheme for fusing relevance values from these methods.

Chapter 3

Relevance Assignment

To analyze input-output relationships in practice, one is presented with discrete data that could be described as a set of M examples (rows of a table) with N variables (columns of a table). Examples represent instances of multiple ground and remotely sensed measurements. Measurements are the variables (features) that might have relationships among themselves. Our goal is to obtain better understanding of the relationships.

In this work, we start with a mathematical definition of variable relevance that is consistent with the conceptual understanding of input-output relationships. We define relevance of an input variable $v_i \in \vec{v} = (v_1, v_2, \ldots, v_N)$ as the partial derivative of an output (predicted) function $f(v_1, v_2, \ldots, v_N)$ with respect to the input variable v_i. Equation 3.1 shows a vector of relevance values for all input variables.

$$\vec{R} = (\frac{\partial f(v_1, v_2, \ldots, v_N)}{\partial v_1}; \ldots; \frac{\partial f(v_1, v_2, \ldots, v_N)}{\partial v_N}) \tag{3.1}$$

This definition assumes that input and output variables are continuous, and an output function f is C^1 continuous (first derivative exists). In order to follow this mathematical definition in practice, there arise challenges associated with (1) processing discrete samples, such as defining the neighborhood proximity of a sample in the manifold of input variables, (2) representing a data-driven model, such as deriving analytical form of a predicted function, (3) scaling input and output variables with different dynamic ranges of measurements, (4) removing dependencies on algorithmic parameters of machine learning techniques, (5) understanding variability of relevance as a function of data quality parameters (e.g., sensor noise, cloud coverage during remote sensing), and (6) treating a mix of continuous and categorical variables, just to name a few.

Our approaches to the above challenges are (a) to perform our analysis on sample partitions obtained using a machine learning technique, (b) to use a mathematically well defined (analytically described) model, like the multi-variate regression, (c) to scale all variables to the same dynamic range of [0, 1], which is discussed in chapter 5, (d) to propose the fusion of multi-method relevance results to increase our confidence in the relevance results, and (e) to investigate dependencies on algorithmic parameters of machine learning techniques and data quality parameters with the experiment results discussed in chapter 6. We verify our methods on data synthesized from analytical functions of various types, as discussed in chapter 5. Having analytical description of a function f allows us to derive relevance according to the definition. We have currently constrained our work to processing only continuous variables and foresee the inclusion of categorical variables in our future work.

Based on the above considerations, we define sample and model relevance assignments for a set of discrete samples with measurements of continuous input and output variables. *Example relevance* R_{ij} is the local relevance of each input variable v_i computed at the sample s_j. The computation is defined for three machine learning techniques in the next sub-sections. *Model relevance* R_i is the global relevance of each input variable v_i over all examples in the entire data-driven model computed by summing all sample relevancies. To obtain comparable example and model relevancies from multiple data-driven models, we normalize the relevance values by the sum of all model relevancies over all input variables (see Equation 3.2). The normalized relevance values are denoted with a tilde.

$$\tilde{R}_{ij} = \frac{R_{ij}}{\sum_i \sum_j R_{ij}}; \tilde{R}_i = \frac{\sum_j R_i}{\sum_i \sum_j R_{ij}} \tag{3.2}$$

In the next subsections, we introduce relevance assignment for Regression Tree (RT), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). The main reason for choosing these three methods comes from our requirement for remote sensing data analysis to process continuous input and output variables. Furthermore, the methods represent a set of machine learning techniques with different underpinning principles for data-driven modeling. The KNN method builds models using only close neighbors to a given sample. The SVM method builds models based on all input samples. As for the RT method, it builds its model by hierarchically subdividing all input samples and fitting a local regression model to samples in each divided cell (leaf). Thus, these methods represent a spectrum of supervised machine learning techniques that would be evaluated in terms of relevance assignment accuracy.

3.1 Regression Tree

The process of building a regression tree based model can be described by splitting input examples into sub-groups (denoted as cells or tree leaves) based on a sequence of criteria imposed on individual variables [10]. The splitting criteria, e.g., information gain or data variance, might depend on problem types, and become one of the algorithmic parameters. In this work, we choose variance as our splitting criterion. Once all examples are partitioned into leaves, a multi-variate linear regression model is used to predict output values for examples that fall into the leaf.

For any example e_j, *example relevance* R_{ij} of the variable v_i is computed from the linear regression model associated with the regression tree leaf. The regression model at each regression tree leaf approximates locally the prediction function with a linear model written as:

$$f(\vec{v}) = \beta_0 + w_1 v_1 + w_2 v_2 + \ldots + w_N v_N \tag{3.3}$$

The example (local) relevance assignment R_{ij} is then computed as a dot product between the normal $\vec{w}$ of a regression model hyper-plane and the unit vector $\vec{u_i}$ of input variable v_i

as described below:

$$R_{ij} = |\vec{w} \bullet \vec{u_i}| \tag{3.4}$$

where $|\vec{w} \bullet \vec{u_i}|$ denotes the absolute value of a dot product of vectors $\vec{w}$ and $\vec{u_i}$.

3.2 K-Nearest Neighbors

K-nearest neighbors is a machine learning method that predicts output values based on K closest examples to any chosen one measured in the space of input variables . Predicted values are formed as a weighted sum of those K nearest examples [16] [10].

For any example e_j, *example relevance* R_{ij} of the variable v_i is computed from the linear regression model obtained from the K nearest neighbors to the example e_j. The linear regression model approximates locally the prediction function f . The example relevance assignment is performed according to Equations 3.3 and 3.4.

3.3 Support Vector Machine

Support vector machine (SVM) is a machine learning method that could build a model for separating examples (classification problem) or for fitting examples (prediction problem). We use SVM as a prediction technique to model input data. The SVM model could be linear or non-linear depending on a SVM kernel. The non-linear models are obtained by mapping input data to a higher dimensional feature space, which is conveniently achieved by kernel mappings [7]. Thus, the prediction function f could be obtained from mathematical descriptions of linear and non-linear kernels. In this thesis, we focus only on a linear model in order to simplify the math.

For a SVM method with the linear kernel, the mathematical description of f becomes a hyper-plane as described in Equation 3.3. The major difference among RT, KNN and SVM methods is in the fact that SVM would use all examples to estimate the parameters of the hyper-plane and it would lead to only one hyper-plane for the entire set of examples. The example relevance assignment for SVM follows Equation 3.4. In the case of SVM, the example relevance and model relevance are identical.

Chapter 4

Relevance Fusion

The goal of relevance fusion is to achieve more robust relevance assignment in the presence of noise, variable data quality (e.g., clouds during remote sensing), as well as to remove any bias introduced by a single machine learning method. The latter motivation is also supported by the no-free-lunch (NFL) theorem [8], which states that no single supervised method is superior over all problem domains; methods can only be superior for particular data sets.

In this thesis, we propose two different schemes to combine the results of relevance assignment, such as average fusion and rank fusion. *Average fusion* is based on taking the numerical average of relevance values, and using it as the combined relevance value. *Rank fusion*, on the other hand, uses the voting scheme to combine the relative ranks of each input variable determined by each machine learning method.

4.1 Average Fusion

Average fusion is executed by taking the normalized relevance results from multiple machine learning methods, and computing the average of them.

Given $\tilde{R}_{ij}^{k}$ as the example relevance of i-th variable at the example e_j from k-th method, the example relevance assignment of $\tilde{R}_{ij}^{avg}$ estabilished using average fusion for the example e_j is defined as :

$$\tilde{R}_{ij}^{avg} = \frac{1}{L} \sum_{k=1}^{L} \tilde{R}_{ij}^{k} \tag{4.1}$$

where L is number of learning methods used.

The model relevance assignment of $\tilde{R}_{i}^{avg}$ using average fusion for an input variable v_i is the sum of all example relevancies as defined in Equation 3.2.

4.2 Rank Fusion

Rank fusion is executed in a similar manner as the average fusion. The difference lies in assigning a relevance rank to each variable v_i based on its relevance ranking by the k-th machine learning model. The absolute magnitude of relevance assignment is lost in rank fusion since the meaning of magnitude is converted to relative ranking during the process. The rank fusion approach is expected to be more robust than the average fusion approach, epsecially when some of the machine learning methods create very incorrect models due to various reasons and skew the correct results of other machine learning methods.

The example relevance assignment using rank fusion is described as follows. For each example e_j and its normalized example relevance $\tilde{R}_{ij}^k$, we define the rank of each variable v_i as the index of a sorted list of relevancies from the smallest to the largest; $rank\{\tilde{R}_{ij}^k\} \subset \{1, 2 \ldots M\}$. The rank fusion based relevance assignment for variable v_i is then computed as shown below:

$$\tilde{R}_{ij}^{rank} = \frac{2}{LN^2} \sum_{k=1}^{L} (N - rank\{\tilde{R}_{ij}^k\}) \tag{4.2}$$

The model relevance assignment of $\tilde{R}_i^{rank}$ using rank fusion for an input variable v_i is the sum of all example relevancies $\tilde{R}_{ij}^{rank}$ over all examples as defined in Equation 3.2.

Chapter 5

Evaluation System Setup

Evaluations were performed with both synthetic and measured data. Synthetic data allow us to simulate three categories of input-output relationships with known ground truth to understand relevance assignment accuracy and relevance dependencies. Measured data was used to demonstrate the application of relevance assignment to study vegetation changes. The results were verified based on our limited understanding of the phenomena.

The next sub-sections describe synthetic data simulations, model building setup, evaluation metrics to assess relevance assignment accuracy, and demonstration of our experimental system.

5.1 Synthetic Data Simulation

Three sets of input-output relationships were simulated to represent (1) linear additive, (2) non-linear additive and (3) non-linear multiplicative categories of relationships. To introduce irrelevant input variables into the problem, we simulated output using only two input variables v_1, v_2 (the relevant variables) while modeling relationships with four variables, where the additional two input variables v_3, v_4 have values drawn from a uniform distribution of [0,1] (the irrelevant variables). The specific analytical forms for generating the three data sets are provided in Equations 5.1 -linear additive, 5.2 -non-linear additive and 5.3 -non-linear multiplicative.

$$f(v_1, v_2, v_3, v_4) = 4v_1 + v_2 \tag{5.1}$$

$$f(v_1, v_2, v_3, v_4) = f(v_1, v_2, v_3, v_4) = \sin \pi v_1 + \cos \frac{\pi}{2} v_2 \tag{5.2}$$

$$f(v_1, v_2, v_3, v_4) = v_1 v_2^2 \tag{5.3}$$

In addition to simulating multiple input-output relationships and relevant-irrelevant variables, we added noise to generated output values to test the noise robustness of relevance assignments. Noise is simulated to be an additive variable following 1D Gaussian distribution with zero mean μ and standard deviation σ; $N(\mu = 0, \sigma)$. The standard deviation was parameterized as $\sigma = \alpha d$, where α is the percentage of the dynamic range d of an output variable. In our experiments, we used $\alpha = 0.1$ and $\alpha = 0.3$ to generate the total of nine synthetic data sets (3 without noise, 3 with additive noise $\alpha = 0.1$, and 3 with additive noise $\alpha = 0.3$).

5.2 Model Building Setup

Model building setup is concerned with optimization of algorithmic parameters and cross validation. First, we set the algorithmic parameters to the following values: (1) RT - variance error as a criterion for splitting, minimum number of examples per leaf to eight, maximum tree depth to 12; (2) KNN - $K = N + 3$ where N is the dimension of all input variables. The reason for setting K slightly larger than the input variable dimensions is to meet the least-square fitting requirements for estimating a hyper-plane from K examples; (3) SVM - linear kernel, cost factor $C = 1.0$, and termination criteria $Epsilon = 0.001$. The optimization of KNN's parameter "K" and RT's parameter "maximum tree depth" was investigated experimentally.

Second, we omitted cross validation of models in our experiments and rather computed input variable relevance based on all available examples. We will investigate in the future the accuracy of input variable relevance assignment from examples selected by cross validation or all available examples.

Finally, KNN and SVM methods are sensitive to the scale of input variables, and will favor variables with a wider scale of values. In order to avoid this type of a bias, the dynamic range of all variables is always normalized to the range between [0, 1] according to the formula below:

$$NormalizedValue = \frac{Value - MinValue}{MaxValue - MinValue} \tag{5.4}$$

5.3 Evaluation Metrics

To evaluate the accuracy of input variable relevance assignment using multiple machine learning methods, we introduced two metrics, such as percentage of correctness (PC) and error distance. The evaluations are conducted only for the synthetic data against the ground truth values of normalized example relevance $\tilde{R}_{ij}^{GT}$ and normalized model relevance $\tilde{R}_{i}^{GT}$. The ground truth values are obtained by computing partial derivatives of Equations 5.1, 5.2 and 5.3 according to Equation 3.1.

5.3.1 Percentage of Correctness

The percentage of correctness metric is defined in Equation 5.5 as:

$$PC = \frac{\sum_{j=1}^{M} \delta_j}{M} \times 100\% \tag{5.5}$$

where δ_j is 1 if

$$\max_i \tilde{R}_{ij}^{GT} = \max_i \tilde{R}_{ij}$$

and is 0 otherwise

5.3.2 Error Distance

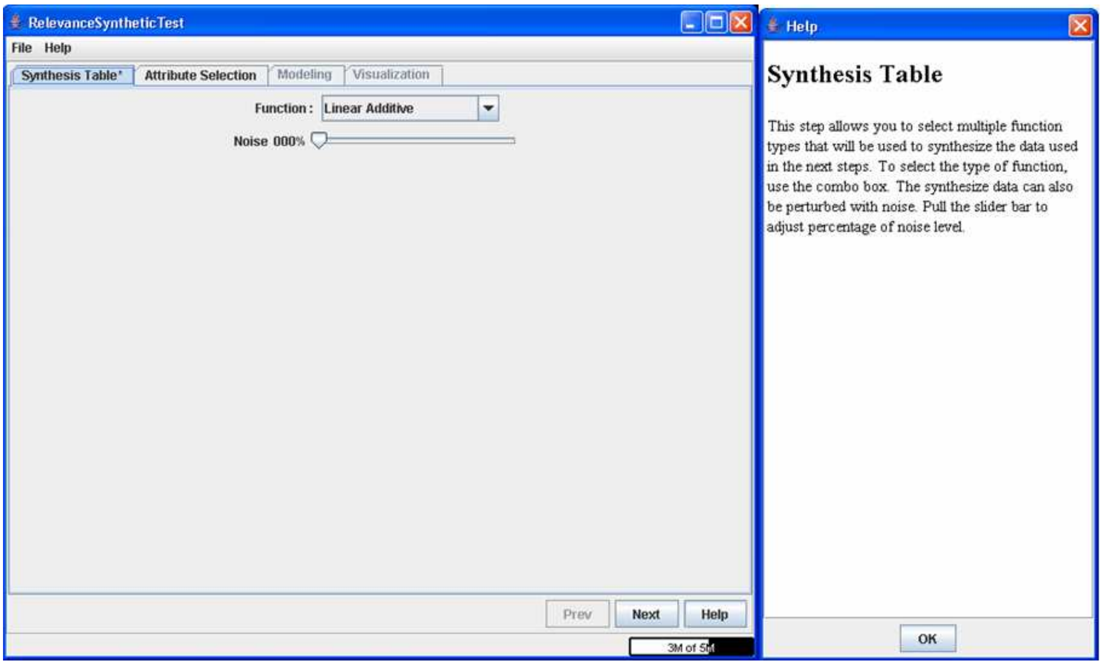
The error distance metric is defined in Equation 5.6 as the Euclidean distance between the true model relevance derived from partial derivative and the relevance estimation from our methods. This metric does not apply to the relevance results obtained using rank fusion since the results are categorical.

$$ErrorDist. = \sum_{i=1}^{N} (\tilde{R}_i^{GT} - \tilde{R}_i)^2 \tag{5.6}$$

5.4 Experimental System

The evaluation of relevance assignments was performed using GeoLearn software that was developed by NCSA and CEE UIUC. GeoLearn allows a user to model input-output variable relationships from multi-variate NASA remote sensing images over a set of boundaries. The machine learning algorithms in the GeoLearn system leverage five software packages, such as, Im2Learn (remote sensing image processing), ArcGIS (georeferencing), D2K software (RT implementation), LibSVM [17] (SVM implementation), and KDTree [16] (KNN implementation).

We integrate these components into a data analysis system for our experiment. The system contains four stages : synthetic parameter setting, attributes selection, learning model selection, and visualization. In the parameter setting stage, the user selects the type of function from the combo box and use the slider bar to adjust the noise level of data, as in Figure 5.1. In the attributes selection stage, the input and output attributes from the data are selected by the user for analysis, as in Figure 5.2. After the user selects the learning method he wish to use in the analysis, the visualization window renders the analysis results such as relevance assignment, data prediction, and prediction error, as in Figure 5.3 and Figure 5.4.



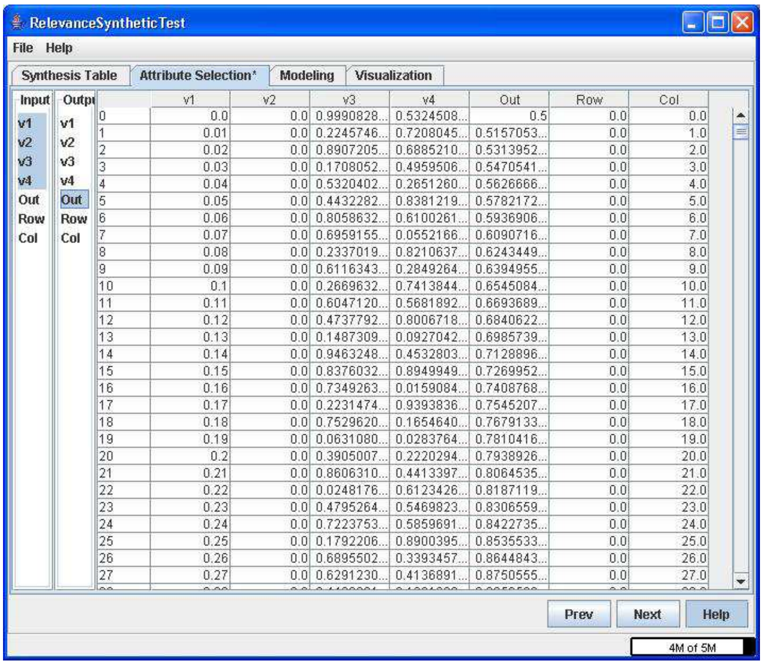
Function : Linear Additive

Noise 000%

Synthesis Table

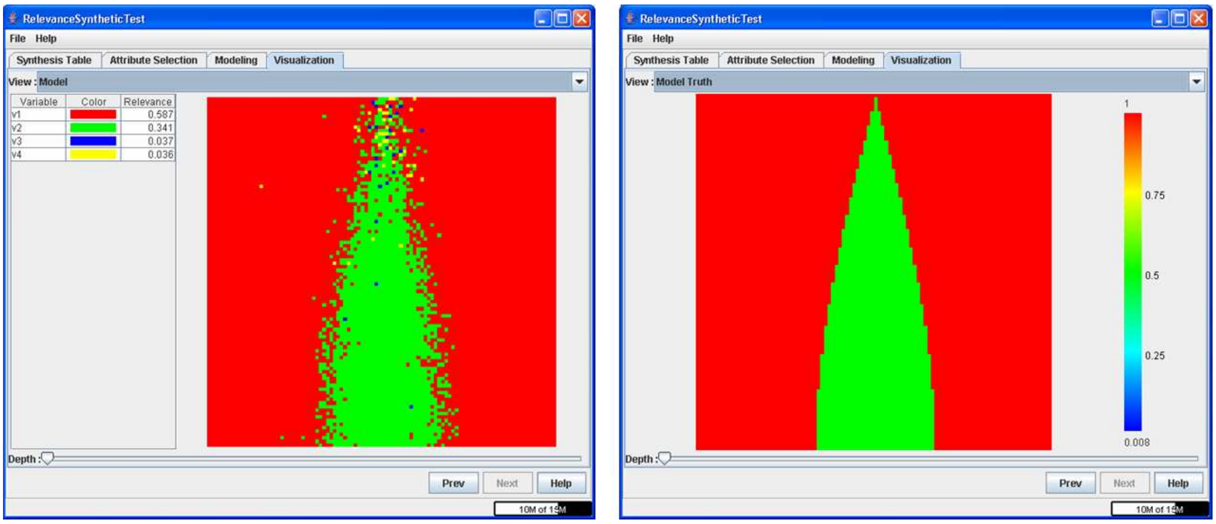
This step allows you to select multiple function types that will be used to synthesize the data used in the next steps. To select the type of function, use the combo box. The synthesize data can also be perturbed with noise. Pull the slider bar to adjust percentage of noise level.

Figure 5.1: Screen capture of our experimental system. The combo box is for selecting the function type, and the slider bar is for changing the noise level. The help screen on the right gives the instruction at each step.



	v1	v2	v3	v4	Out	Row	Col
0	0.0	0.0	0.9990828...	0.5324508...	0.5	0.0	0.0
1	0.01	0.0	0.2245746...	0.7208045...	0.5157053...	0.0	1.0
2	0.02	0.0	0.8907205...	0.6885210...	0.5313952...	0.0	2.0
3	0.03	0.0	0.1708052...	0.4959506...	0.5470541...	0.0	3.0
4	0.04	0.0	0.5320402...	0.2651260...	0.5626666...	0.0	4.0
5	0.05	0.0	0.4432282...	0.8381219...	0.5782172...	0.0	5.0
6	0.06	0.0	0.8058632...	0.6100261...	0.5936906...	0.0	6.0
7	0.07	0.0	0.6959155...	0.0552166...	0.6090716...	0.0	7.0
8	0.08	0.0	0.2337019...	0.8210637...	0.6243449...	0.0	8.0
9	0.09	0.0	0.6116343...	0.2849264...	0.6394955...	0.0	9.0
10	0.1	0.0	0.2669632...	0.7413844...	0.6545084...	0.0	10.0
11	0.11	0.0	0.6047120...	0.5681892...	0.6693689...	0.0	11.0
12	0.12	0.0	0.4737792...	0.8006718...	0.6840622...	0.0	12.0
13	0.13	0.0	0.1487309...	0.0927042...	0.6985739...	0.0	13.0
14	0.14	0.0	0.9463248...	0.4532803...	0.7128896...	0.0	14.0
15	0.15	0.0	0.8376032...	0.8949949...	0.7269952...	0.0	15.0
16	0.16	0.0	0.7349263...	0.0159084...	0.7408768...	0.0	16.0
17	0.17	0.0	0.2231474...	0.9393836...	0.7545207...	0.0	17.0
18	0.18	0.0	0.7529620...	0.1654640...	0.7679133...	0.0	18.0
19	0.19	0.0	0.0631080...	0.0283764...	0.7810416...	0.0	19.0
20	0.2	0.0	0.3905007...	0.2220294...	0.7938926...	0.0	20.0
21	0.21	0.0	0.8606310...	0.4413397...	0.8064535...	0.0	21.0
22	0.22	0.0	0.0248176...	0.6123426...	0.8187119...	0.0	22.0
23	0.23	0.0	0.4795264...	0.5469823...	0.8306559...	0.0	23.0
24	0.24	0.0	0.7223753...	0.5859691...	0.8422735...	0.0	24.0
25	0.25	0.0	0.1792206...	0.8900395...	0.8535533...	0.0	25.0
26	0.26	0.0	0.6895502...	0.3393457...	0.8644843...	0.0	26.0
27	0.27	0.0	0.6291230...	0.4136891...	0.8750555...	0.0	27.0

Figure 5.2: Screen capture of our experimental system. The attributes in the input and output columns are selected for analysis of input/output relationships.

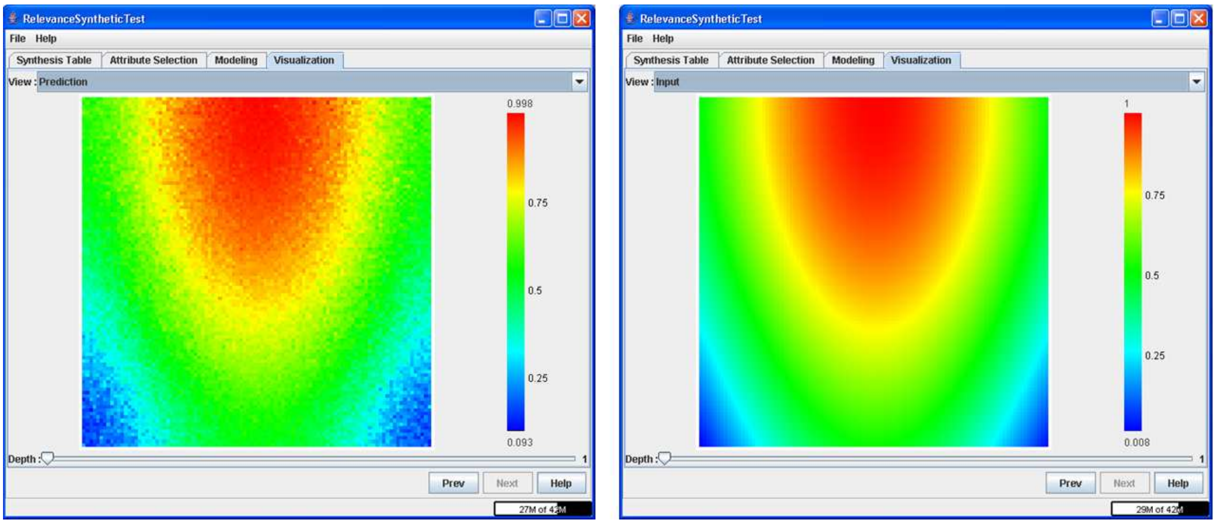


View : Model

Variable	Color	Relevance
v1	🟥	0.587
v2	🟩	0.341
v3	🟦	0.037
v4	🟨	0.036

View : Model Truth

Figure 5.3: Screen capture of our experimental system. The output image visualizes the relevance assignment results. The user can use the combo box to select different option of visualization. Left : predicted relevance assignment from KNN model. Right : ground truth relevance assignment.



View : Prediction

View : Input

Figure 5.4: Screen capture of our experimental system. The output image visualizes the output prediction results using KNN method. Left : predicted values from KNN model. Right : ground truth values of data.

Chapter 6

Experiment Results

In this section, we present evaluations with synthetic and measured data in two forms. First, we report a *relevance image* that shows the color of an input variable with maximum relevance value at each pixel location. The color coding schema maps red, green, blue and yellow colors to (v_1, v_2, v_3, v_4). Second, we provide a *relevance table* with model relevance value for each input variable.

6.1 Synthetic Data

6.1.1 Relevance Assignments Results

The relevance assignment results using RT, KNN, and SVM methods from synthetic data are summarized in Figure 6.1 and Table 6.1. As we can see from the results, SVM is very robust for linear data, while RT and KNN perform better for non-linear data.

6.1.2 Relevance Fusion Results

The results obtained using fusion methods for the synthetic data are summarized in Figure 6.2 and Table 6.2. Both fusion schemes perform reasonably well by considering relevance results from multiple methods. However, when the result from one of these methods goes very wrong, the averaging scheme can sometimes be affected and fail to give a correct result. As we can see in Figure 6.2 and Table 6.2, the results of averag fusion are affected due to the poor performance of SVM for non-linear additive data.

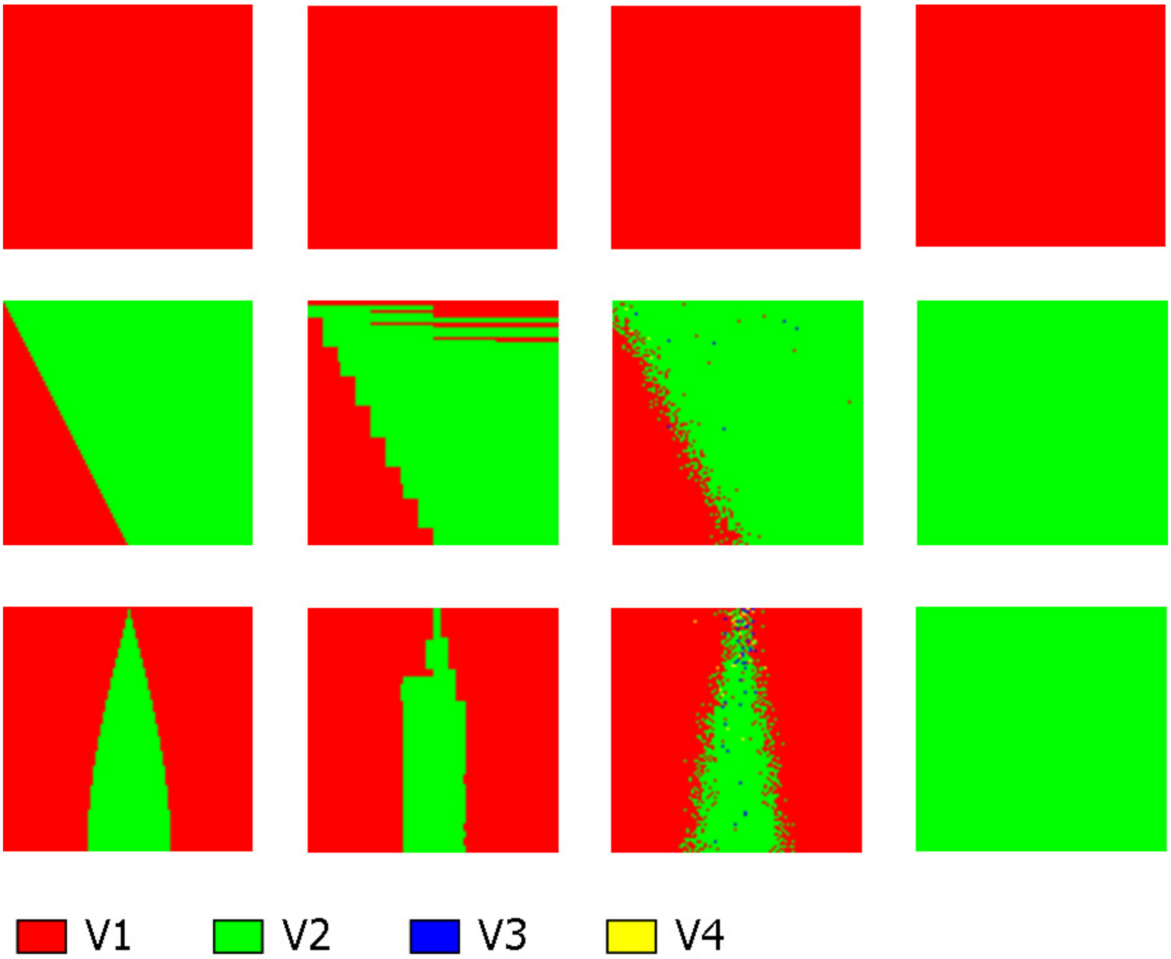


Figure 6.1: Relevance assignment images for synthetic data. From top row to bottom row : linear data, non-linear multiplicative data, and non-linear additive data. From left to right : ground truth, regression tree, k-nearest neighbors, and support vector machine.

Model	Var	Linear		Non-Linear Add.		Non-Linear Mul.	
		Relevance	Correct Percent	Relevance	Correct Percent	Relevance	Correct Percent
			Error Dist.		Error Dist.		Error Dist.
RT	v1	0.8	100%	0.667415	95.5%	0.423898	91.83%
	v2	0.2		0.330931		0.575511	
	v3	3.0E-17	4.0E-29	8.54E-4	2.79E-6	2.73E-4	0.001
	v4	3.72E-17		8.01E-4		3.17E-4	
KNN	v1	0.8	100%	0.616814	95.07%	0.35364	96.94%
	v2	0.2		0.326967		0.568512	
	v3	7.48E-14	2.24E-25	0.027992	0.004263	0.039011	0.00621
	v4	3.53E-13		0.028227		0.038838	
SVM	v1	0.748753	100%	0.0.040722	20.69%	0.391619	75%
	v2	0.249007		0.95396		0.598363	
	v3	6.29E-4	0.00503	0.005292	0.781345	0.001454	0.000168
	v4	0.00161		2.69E-5		0.008564	

Table 6.1: Relevance assignment results using regression tree, K-nearest neighbors, and support vector machine.

Model	Var	Linear		Non-Linear Add.		Non-Linear Mul.	
		Relevance	Correct Percent	Relevance	Correct Percent	Relevance	Correct Percent
			Error Dist.		Error Dist.		Error Dist.
Fusion Avg	v1	0.782918	100%	0.44165	68.67%	0.389719	97.72%
	v2	0.216336		0.537286		0.580795	
	v3	2.0E-4	0.000558	0.011379	0.093948	0.013579	0.001
	v4	5.37E-4		0.009685		0.015906	
Fusion Rank	v1	0.8	100%	0.3	97.58%	0.3	97.62%
	v2	0.2		0.4		0.4	
	v3	7.48E-14	NA	0.133333	NA	0.166667	NA
	v4	3.53E-13		0.166667		0.133333	

Table 6.2: Relevance assignment results using fusion schemes.

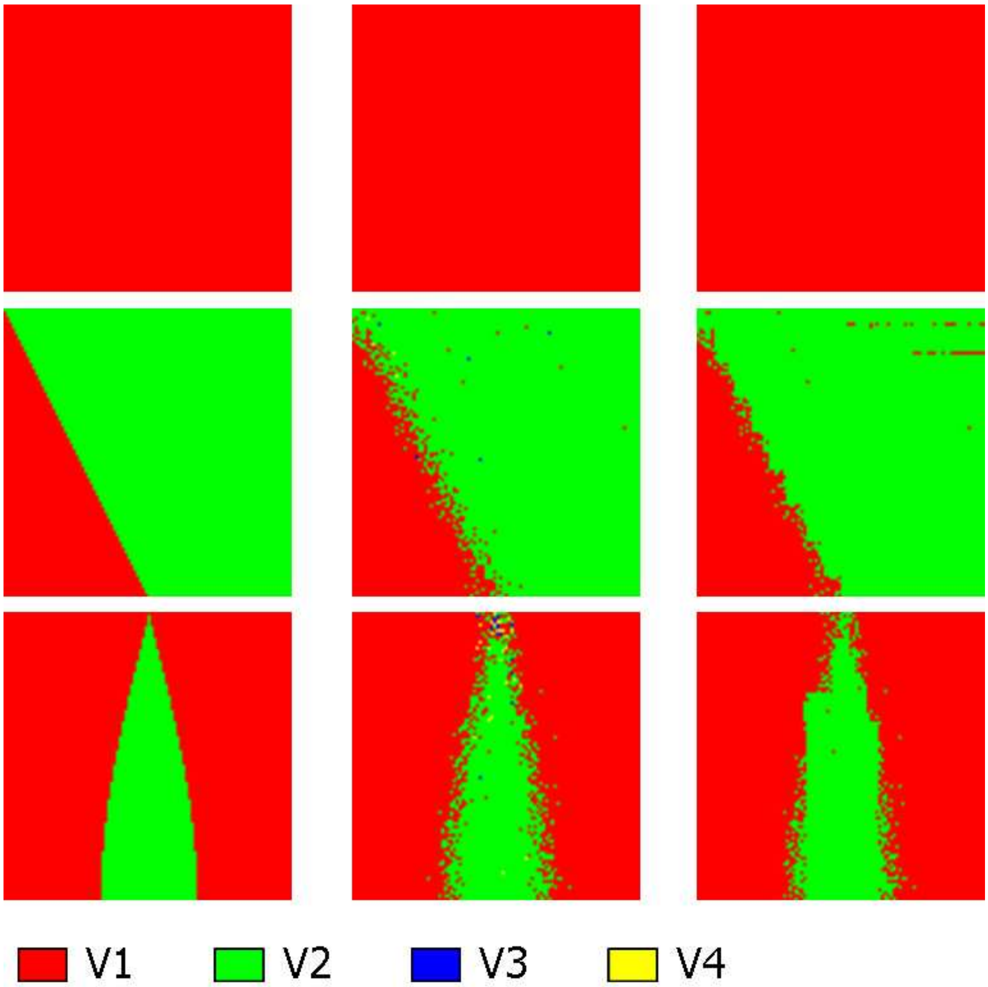


Figure 6.2: Relevance assignment images for synthetic data using relevance fusion. From top row to bottom row : linear data, non-linear multiplicative data, and non-linear additive data. From left to right : ground truth, average fusion, and rank fusion.

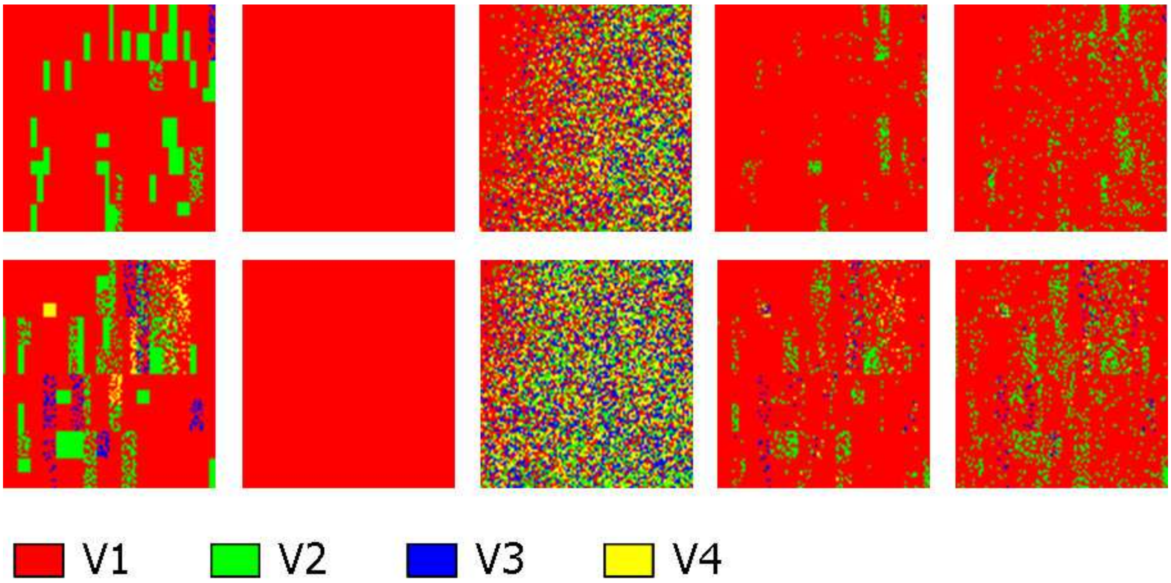


Figure 6.3: Relevance assignment images for linear data with noise. Top row : linear data with 10% noise. Bottom row : linear data with 30% noise. From left to right : regression tree, support vector machine, k-nearest neighbors, average fusion, and rank fusion.

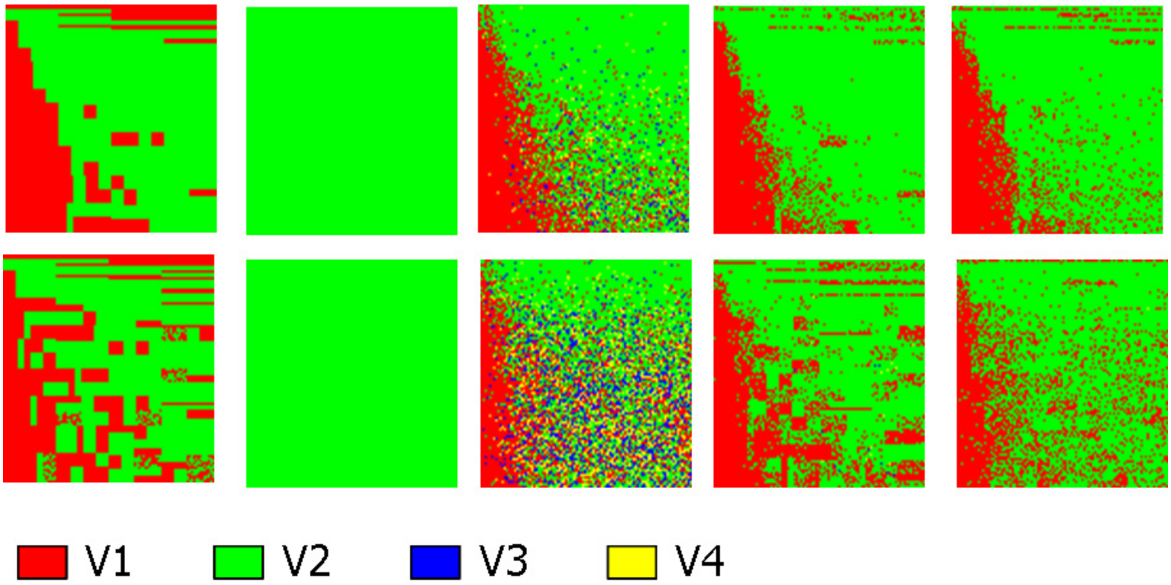


Figure 6.4: Relevance assignment images for non-linear multiplicative data with noise. Top row : data with 10% noise. Bottom row : data with 30% noise. From left to right : regression tree, support vector machine, k-nearest neighbors, average fusion, and rank fusion.

Model	Var	Linear Data Add. : Relevance with Noise (%)					
		10 %	Correct Percent	Error Dist.	30 %	Correct Percent	Error Dist.
RT	v1	0.711588	84.29%	0.01062	0.644956	86.15%	0.033657
	v2	0.240934			0.22233		
	v3	0.024482			0.069678		
	v4	0.022996			0.063036		
KNN	v1	0.397197	45.59%	0.2378252	0.288214	86.15%	0.3753629
	v2	0.214593			0.238612		
	v3	0.192119			0.237003		
	v4	0.196092			0.236171		
SVM	v1	0.788641	100%	0.0001776	0.79588	100%	2.46E-5
	v2	0.201972			0.200412		
	v3	0.005282			0.002392		
	v4	0.004105			0.001317		
Fusion Avg	v1	0.632475	96.88%	0.0394371	0.57635	91.74%	0.07108
	v2	0.219166			0.220451		
	v3	0.073961			0.103024		
	v4	0.074398			0.100174		
Fusion Rank	v1	0.4	91.18%	NA	0.4	89.11%	NA
	v2	0.3			0.2		
	v3	0.133333			0.266667		
	v4	0.166667			0.133333		

Table 6.3: Summary of relevance assignments for linear synthetic data with noise

Model	Var	Non-Linear Data Mul. : Relevance with Noise (%)					
		10 %	Correct Percent	Error Dist.	30 %	Correct Percent	Error Dist.
RT	v1	0.405985	84.93%	0.000593	0.436787	72.75%	0.01111
	v2	0.577826			0.507675		
	v3	0.008041			0.027279		
	v4	0.008148			0.028259		
KNN	v1	0.303406	82.41%	0.05893	0.272351	56.16%	0.157361
	v2	0.455939			0.3415		
	v3	0.119827			0.191977		
	v4	0.120828			0.194172		
SVM	v1	0.395111	75%	5.02E-5	0.414512	75%	0.001425
	v2	0.601205			0.566833		
	v3	0.001396			0.004211		
	v4	0.002289			0.014444		
Fusion Avg	v1	0.368167	90.14%	0.00893	0.37455	77.67%	0.028569
	v2	0.54499			0.472003		
	v3	0.043088			0.074489		
	v4	0.043755			0.078959		
Fusion Rank	v1	0.333333	88.37%	NA	0.333333	76.41%	NA
	v2	0.366667			0.366667		
	v3	0.166667			0.166667		
	v4	0.133333			0.133333		

Table 6.4: Summary of relevance assignments for non-linear multiplicative synthetic data with noise

		Non-Linear Data Add. : Relevance with Noise (%)					
Model	Var	10 %	Correct Percent	Error Dist.	30 %	Correct Percent	Error Dist.
RT	v1	0.565207	68.44%	0.014285	0.463235	58.57%	0.056862
	v2	0.372952			0.378203		
	v3	0.033536			0.082852		
	v4	0.028305			0.07571		
KNN	v1	0.353553	53.03%	0.018587	0.275271	31.87%	0.274157
	v2	0.247346			0.249596		
	v3	0.200375			0.239165		
	v4	0.198726			0.235968		
SVM	v1	0.040941	20.69%	744761	0.025481	20.69%	0.805613
	v2	0.923234			0.957779		
	v3	0.031645			0.005078		
	v4	0.00418			0.011662		
Fusion Avg	v1	0.3199	57.14%	0.168686	0.254662	42.32%	0.233466
	v2	0.514511			0.528526		
	v3	0.088518			0.109032		
	v4	0.07707			0.10778		
Fusion Rank	v1	0.366667	64.85%	NA	0.366667	50.74%	NA
	v2	0.333333			0.333333		
	v3	0.2			0.133333		
	v4	0.1			0.166667		

Table 6.5: Summary of relevance assignments for non-linear additive synthetic data with noise

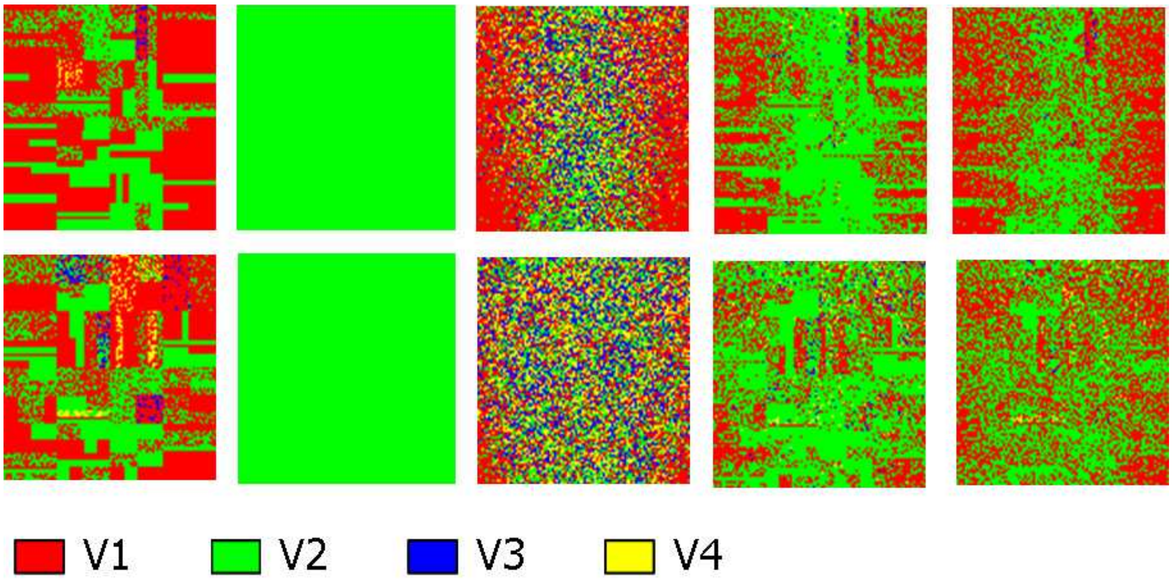


Figure 6.5: Relevance assignment images for non-linear additive data with noise. Top row : data with 10% noise. Bottom row : data with 30% noise. From left to right : regression tree, support vector machine, k-nearest neighbors, average fusion, and rank fusion.

6.1.3 Results for Synthetic Data with Noise

We also evalutate the methods with the same synthetic data with noise added. The resulting relevance assignments are summarized in Table 6.3, Table 6.4, and Table 6.5. The relevance images are shown in Figure 6.3, Figure 6.4, and Figure 6.5 for each set of data. The performances from our methods are affected by the added noise in different degrees. From these results, we see that the K-nearest neighbors are more vulnerable to noise than other methods, and support vector machine is relatively more robust for it limited degree of freedom.

6.2 Measured Data

We processed remotely sensed data from NASA acquired in 2003, at spatial resolution 1000m per pixel and at the location (latitude x longitude) = ([35.34, 36.35] x [-91.54, -93.32]. We model the ouput variable Fpar (fraction of photosynthetic active radiation) as

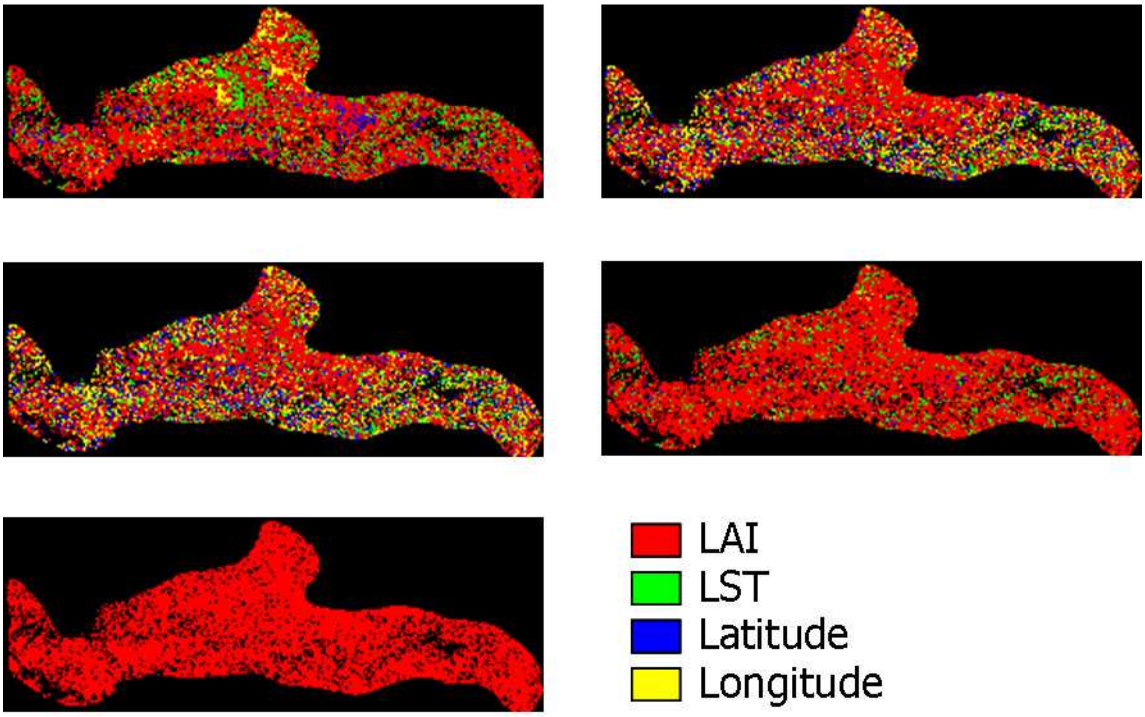


Figure 6.6: Relevance images for measured data. Left column from top to bottom : regression tree, k-nearest neighbors, and support vector machine. Right column from top to bottom : average fusion and rank fusion. The most relevant variable at each pixel is indicated by color.

Model	Var	Relevance
RT	LAI	0.996014
	LST	0.003058
	Latitude	1.07E-4
	Longitude	8.20E-4
KNN	LAI	0.449047
	LST	0.178427
	Latitude	0.17511
	Longitude	0.197416
SVM	LAI	0.885942
	LST	0.073454
	Latitude	0.040576
	Longitude	2.82E-5
Fusion Avg	LAI	0.777001
	LST	0.08498
	Latitude	0.071931
	Longitude	0.066088
Fusion Rank	LAI	0.4
	LST	0.266667
	Latitude	0.133333
	Longitude	0.2

Table 6.6: Relevance assignment results for measured data provided by NASA.

a function of input variables consisting of LST (land surface temperature), LAI (leaf area index), Latitude, and Longitude. Though we do not have the underlying analytical model for the relationship among these input-output variables, we anticipated that for this geospatial location, Fpar is more relevant to LAI, and both Latitude and Longitude are not relevant to Fpar. Therefore, the expected input-output variable relationship is the relevance of LAI in terms of output variable Fpar should be the largest value and close to 1, while the relevance of LST should be much smaller. The Latitude and Longitude should be considered as irrelevant noise, with relevance values close to 0. The relevance results are summarized in Figure 6.6 and Table 6.6. As we can see from the results, all of our proposed methods yield expected results by predicting LAI as the most relevant input variable in predicting Fpar. Based on the experiment, we get consistent results from these methods, and they are similar to the results we have from linear synthetic data with noise added. Therefore, we are confident that the data relationship can be approximated by the linear model with noise in measurement.

6.3 Relevance Assignment As a Function of Algorithmic Parameters

In this test, we would like to understand the variability of relevance assignment as the function of different algorithm parameters. Here we test the tree depth parameter for RT, and the number K neighbors for KNN. For each data set, we use different tree depth for the same regression tree model trained from the data and compute the correctness of relevance assignment. A similar test is also performed for KNN algorithm using different K parameters. In our experiment, we set the range of tree depth between 3 to 10, and the range of K parameter between M+3 to M+303, where M is the input dimensions. The test dataset are the non-linear multiplicative data and non-linear additive data defined in chapter4. We conveniently skip the linear data because both RT and KNN methods will locally approximate the data with linear regression. Therefore, they will yield approximately identical relevance assignment results for linear data, regardless of the different parameters we set.

The results of the experiments are plotted in Figure 6.7 and Figure 6.8. For regression tree, we observed a similar trend here as from the results of previous work by White et al[24]. In their work, they observed that as the tree depth increase to certain point, the testing error start to increase, which is due to the over-fitting of the model to data. Similarly, as we increase the tree depth, we also observed that after a particular depth, the error of relevance assignments start to increase. For KNN, we observed similar trends for opposite reason. As we increase the number of K, the closest K points cover a larger region of input space. Therefore after passing certain threshold, the linear regression model computed from these K points does not have enough expressivity for the variation of this large region. Thus the relevance assignment becomes inaccurate for this model.

If we see both methods in a more unified way, they both perform the partition of examples. RT subdivides the input space into a tree hierarchy, and KNN selects the K neighbors in its proximity. The different tree depth parameter and the K neighbors parameter both give clue about the size of examples in the partition for our relevance assignment method. Assuming there are M input examples, the approximate number of examples for each leaf node will be $\frac{M}{2^d}$, where d is the depth of tree. We can therefore relate the K parameter in KNN to the tree depth parameter in RT, using the formula :

$$K = \frac{M}{2^d} \tag{6.1}$$

and

$$d = \log \frac{M}{K} \tag{6.2}$$

where d is the depth of regression tree, and K is the K neighbors parameter. From our experiment, we see that the peak of the correctness for RT and KNN have their parameters following the relationship in Equation 6.2. Therefore we can utilize this parameter

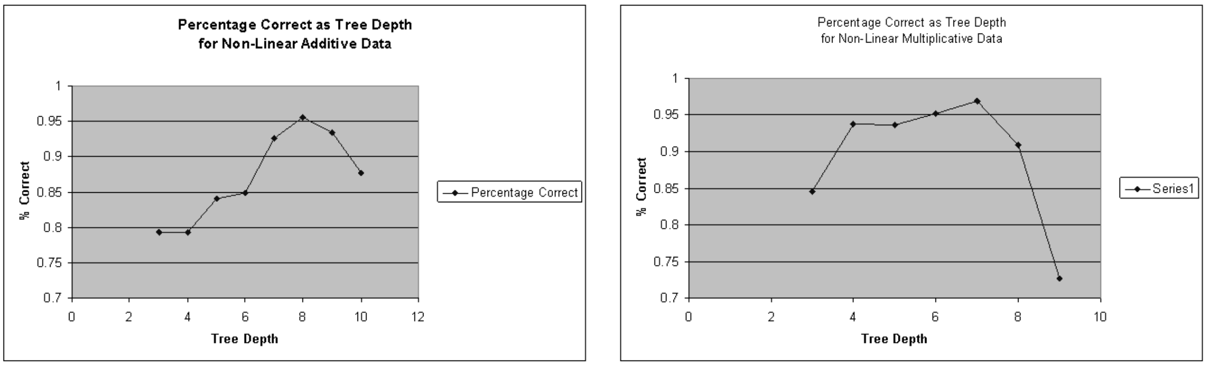


Figure 6.7: Plot of percentage correct as the function of regression tree depth. (Left : Non-linear additive data. Right : Non-linear multiplicativedata.)

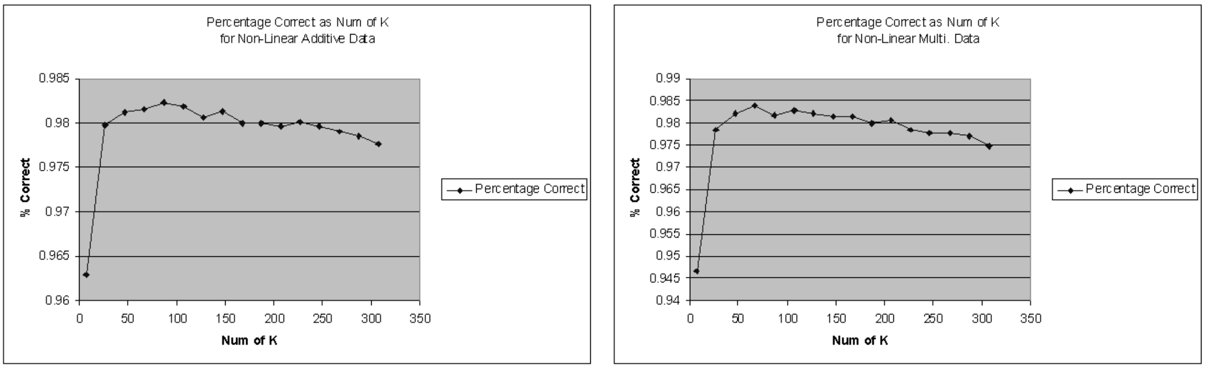


Figure 6.8: Plot of percentage correct as the function of parameter K in KNN. (Left : Non-linear additive data. Right : Non-linear multiplicativedata data.)

relationship to optimize the performance of our relevance assignment method.

6.4 Relevance Assignment As a Function of Data Quality

In this experiment, we wish to understand the results of relevance assignment as a function of data quality. We test our relevance assignment methods on the measured data with different level of data quality. The remote sensing data provided by NASA come with the quality control mask associated with each input variables. By applying different quality criteria (QA masks), we can either discard the unreliable measurements (due to cloud covering,

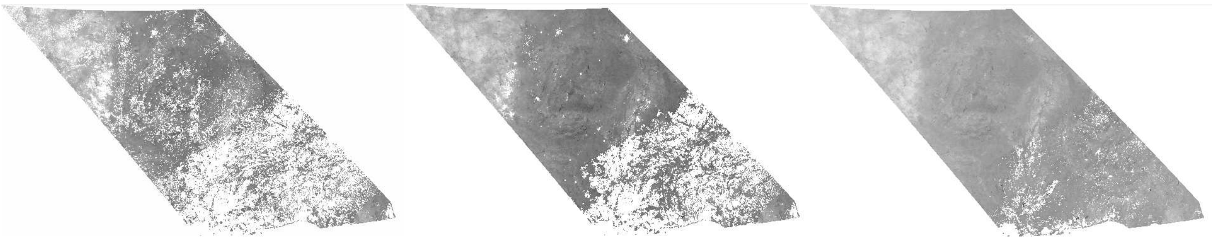


Figure 6.9: The images generated from measured data by applying quality control masks of different levels. The white area indicate pixels removed after applying the mask. From left to right : high quality data, medium quality data, and low quality data.

sensor noise, and etc), or process measurements.

For our experiments, we use the same input-output variables from the same geographic location as in the previous section. We build three different test datasets representing high quality, medium quality, and low quality data using different quality bit setting. There are total of eight quality control parameters for the input data. Among them, four are used for LST variable and the other four are for LAI/Fpar variables. Each of them has the discrete value from 0 to 3, where 0 represents the highest quality, and 3 represents the lowest quality. Although there are totally 2^8 possible combinations for all of these control parameters, we observed that most of the combinations yield the same data. For simplicity, we chose only those three of the combinations which led to significantly different data. We labeled them as high, medium, and low quality data according to the number of high quality bits in the quality control parameters. The data labeled with high quality should have the most of its parameters set to 0 (highest quality), while the data labeled with low quality should have very few of the parameters set to high quality. The settings of quality control parameters for the high, medium, and low quality data are summarized in Table 6.7. The results after applying different quality control masks are shown in Figure 6.9. The total numbers of examples after QA processing are 4,286 examples for high quality data, 10,502 examples for medium quality data, 12,433 examples for low quality data.

The relevance assignment results for the datasets are summarized in Table 6.8. The average relevance of LAI from different quality data is shown in the chart of Figure 6.10.

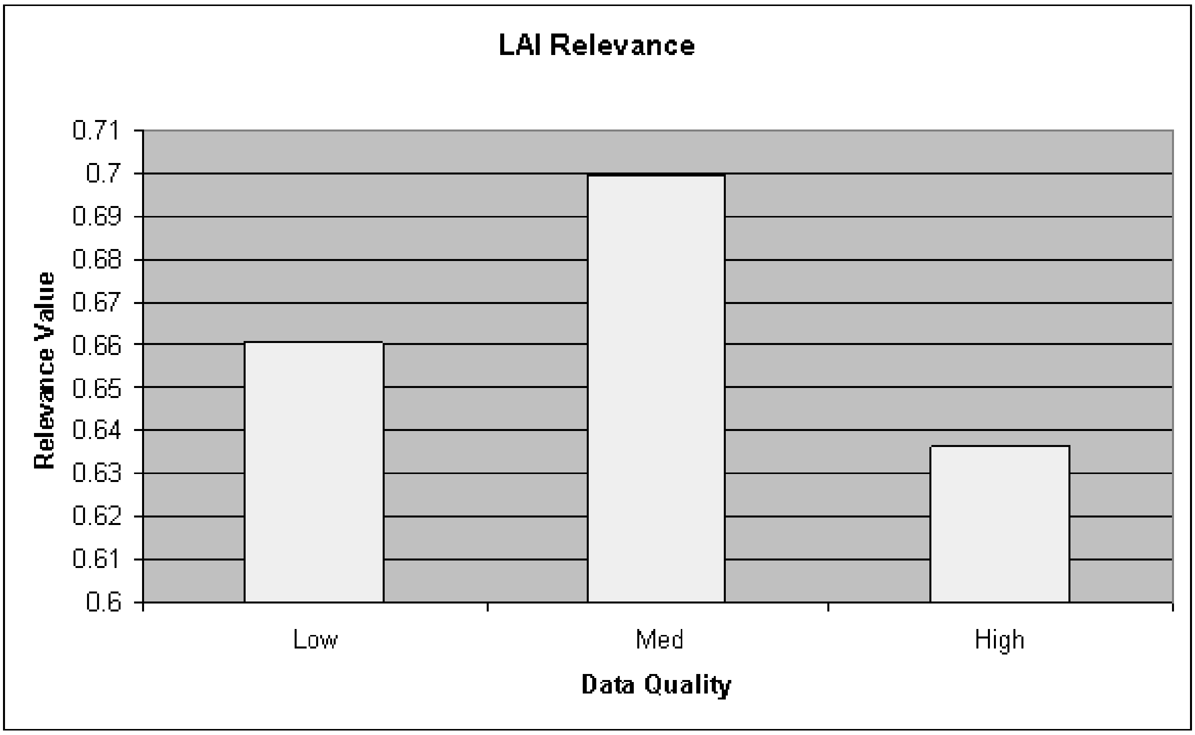


Figure 6.10: Chart of LAI relevance as the function of data quality.

The reason for selecting LAI for observation is that from previous experiments in measured data, we found LAI as the most relevant variable for predicting Fpar. Thus we expect to understand the change in relevance assignment through the observation of LAI variable. We hypothesized that the relevance of LAI would increase with the higher quality of input data. However, the results did not confirm our hypothesis. We believe that the relevance value might vary within 7% as a function of data quality, assuming that statistically sufficient number of examples are processed.

Variable	QA Control Bit	Parameter Quality Settings		
		High	Medium	Low
LST	Mandatory QA	0	3	3
	Data Quality	3	3	3
	Emissivity Error	0	0	3
	LST Error	0	0	3
LAI/Fpar	MODLAND QA	1	1	3
	Algorithm Path	0	0	0
	Landcover Source	0	0	0
	Pixel Quality	0	3	3

Table 6.7: The parameter settings for QA control bits. All QA control bits have the discrete range from 0 to 3, where 0 denotes the highest quality and 3 denotes the lowest quality. We choose 3 combinations of these data quality bits to represent data of high, medium, and low qualities. Other combinations are omitted because they do not yield significant difference from the data generated with the chosen combinations.

Model	Var	Relevance		
		High Quality	Medium Quality	Low Quality
RT	LAI	0.633368	0.754668	0.705934
	LST	0.149101	0.099970	0.103617
	Latitude	0.101862	0.065668	0.078714
	Longitude	0.115667	0.079693	0.111734
KNN	LAI	0.395249	0.520158	0.517823
	LST	0.179153	0.145516	0.144572
	Latitude	0.191304	0.153562	0.154405
	Longitude	0.234292	0.180762	0.183198
SVM	LAI	0.880259	0.823931	0.757232
	LST	0.094106	0.140822	0.197846
	Latitude	0.011883	0.023401	0.026861
	Longitude	0.013750	0.011845	0.018059
Fusion Avg	LAI	0.636292	0.699585	0.660330
	LST	0.140787	0.128769	0.148678
	Latitude	0.101683	0.080877	0.086660
	Longitude	0.121236	0.090767	0.104330
Fusion Rank	LAI	0.4	0.4	0.4
	LST	0.233333	0.233333	0.2
	Latitude	0.133333	0.166667	0.166667
	Longitude	0.233333	0.2	0.233333

Table 6.8: Results of variability test for relevance assignment using measured data under different level of quality. The mandatory QA bit is set to 1 for high quality data, and is 3 for low quality data. The total numbers of examples after processing are 4,286 examples for high quality data, 10,502 examples for medium quality data, 12,433 examples for low quality data.

Chapter 7

Discussion

The results presented in the previous chapter are discussed by comparing individual machine learning methods and fusion methods.

7.1 Comparison of Individual Methods

For linear data without noise, all of our methods perform well and give accurate results in both percentage of correctness (PC), and error distance metrics. However, for non-linear data, SVM constantly gives poor performance. This is due to the fact that we use linear kernel, and therefore, its expressivity is limited. The performance drop is especially drastic when we test it with non-linear additive data, when there are periodic changes of the data. Though both RT and KNN perform relatively well under non-linear dataset, we observe some noisy pattern for the relevance image from KNN. This is because KNN is a local method only considering its close neighbors, and is more sensitive to noise of the data. On the other hand, RT is more robust to noise of data but usually cannot give a very accurate approximation of the relevance image. This is because RT works by subdividing the input space into individual cells, and assigns a prediction model for each cell. Therefore, its accuracy is limited by the level of this subdivision, or in other words, the tree depth. Though increasing the maximum tree depth might give us a better relevance approximation, in reality it is usually preferred to grow the tree only to a certain depth and not to over-fit the data.

The robustness properties mentioned above are especially obvious under noisy data. As we can see from Figure 6.3, Figure 6.4, and Figure 6.5, the correctness of relevance assignments for KNN are affected by the noise in data. SVM, on the other hand, benefits from its lack of degree of freedom. The performance drop under noise is relatively small for SVM with linear kernel. We also note an interesting result when testing non-linear multiplicative data under noise, where SVM outperforms other methods. Since the original data is monotonically increasing, the linear kernel happens to be a moderate approximation for it. Therefore, when testing with the same data with noise added, SVM is able to yield a better result than RT and KNN.

In summary, RT is a reliable hybrid method that usually gives accurate relevance estimation. KNN is flexible under different data type and always gives good relevance estimation in terms of correctness. However, it is very sensitive to noise, and we observe significant performance drops when noise presents. SVM usually yields more robust result under noise, but is restricted by its expressivity since we use only linear kernel here. There is no single best method for every dataset. The correctness of relevance assignment strongly depends on the type of data and the learning method we use. The fusion methods are proposed based on this motivation.

7.2 Comparison of Fusion Methods

The goal of relevance fusion is to take advantage of each method and to increase the stability of relevance assignment results. From experiments, fusion methods usually outperform any single method in terms of relevance assignment correctness (PC). For data without noise, the difference between average fusion and rank fusion is not obvious. In terms of error distance, average fusion is a better choice than rank fusion scheme, since it preserves the relevance value magnitude. However, since average fusion takes the average of relevance values across different methods, the error distance it obtains will not be optimal.

For dataset with noise, the correctness of relevance assignment for fusion methods is stable, though sometimes not optimal comparing to the best results from all the individual method. However, it is still desirable to use fusion method since they perform more stable under various data type and noise setting.

The two schemes usually work equally well in most circumstances in terms of correctness. However, there are situations the rank fusion scheme might yield a more reliable result. One of the examples is for non-linear additive data, when SVM gives a very poor estimation of relevance assignment. In this case, the performance of averaging scheme will be pulled down by this poor result, and cannot yield accurate results comparing to ranking scheme (See Figure 6.2 and Table 6.2).

7.3 Application to Real World Data

For real measured remotely sensed data, our experiments show consistent results of the relevance assignments. The ranking of relevance for input variables from high to low are LAI, LST, Latitude and Longitude. The amplitude of relevance estimation results indicate that LAI is much more relevant than other variables in predicting Fpar. Though all of the methods give us consistent relevance assignment, result from KNN is is less reliable comparing to other methods. The relevance assignment obtained from KNN is not as discriminative for LAI to other variables. We believe this is due to the fact that KNN is more sensitive to noise, and its relevance assignments are therefore affected by noise in the data measurement.

Chapter 8

Conclusion

In this thesis, we proposed a framework for computing input variable relevance with respect to a predicted output variable from multiple machine learning methods. Following the conceptual definition of relevance in the literature, we defined partial derivatives of input-output dependencies as our relevance assignment approach. The estimation of two types of relevancies, such as example and model relevancies, were implemented for regression tree, K-nearest neighbors, and support vector machine methods. Additional fusion schemes for combining the relevance results from multiple methods were evaluated together with single methods by using synthetic and measured data and two metrics. Based on three categories of synthetic input-output simulations including linear additive, non-linear additive and non-linear multiplicative relationships without or with noise added, we concluded that the relevance assignment using fusion approaches demonstrate more robust performance than the assignment using a single machine learning method.

In the future, we would like to extend the fusion methods to include the results from other learning methods, and to understand the dependencies of relevance assignment on model building setups.

References

[1] E. Alpaydin. *Introduction to Machine Learning.* The MIT Press, 2004.

[2] S. De Backer, P. Kempeneers, W. Debruyn, and P. Scheunders. A band selection technique for spectral classification. *IEEE Geoscience and Remote Sensing Letters*, 2(3):319–323, July 2005.

[3] P. Bajcsy and P. Groves. Methodology for hyperspectral band selection. *Photogrammetric Engineering and Remote Sensing Journal*, 70(7):793–802, 2004.

[4] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19:711–720, 1997.

[5] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 1997.

[6] I. Cohen, Q. Tian, X. Zhou, and T. Huang. Feature selection using principal feature analysis. *submitted to ICIP'02*, 2002.

[7] N. Cristianini and J.S. Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, 2000.

[8] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* Wiley-Interscience, 2000.

[9] P. Groves and P. Bajcsy. Methodology for hyperspectral band and classification model selection. *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data*, 2003.

[10] J. Han and M. Kamber. *Data Mining : Concepts and Techniques.* Morgan kaufmann Publishers, 2001.

[11] M. Heiler, D. Cremers, and C. Schnorr. Efficient feature subset selection for support vector machines. Technical Report 21, Dept. of Math and Computer Science, Universitty of Mannheim, 2001.

[12] A. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Machine Intell.*, 19:153–189, 1997.

[13] I.T. Jolliffe. *Principal Component Analysis.* Springer, 2002.

[14] P. Kempeneers, S. De Backer, W. Debruyn, P. Coppin, and P. Scheunders. Generic wavelet based hyperspectral classification applied to vegetation stress detection. *IEEE Transaction on Geoscience and Remote Sensing*, 2(3):319–323, July 2005.

[15] M. Kudo and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33:25–41, 2000.

[16] S. D. Levy. A java class for kd-tree search. http://www.cs.wlu.edu/∼levy.

[17] C. J. Lin. Libsvm. http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[18] G.P. McCabe. Principal variables. *Technometrics*, 26:127–134, 1994.

[19] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition.* Wiley-Interscience, 2004.

[20] P. Perner. Improving the accuracy of decision tree induction by feature pre-selection. *Applied Artificial Intelligence*, 15(8):747–760, 2001.

[21] P. Perner and C. Apte. Empirical evaluation of feature subset selection based on a real world data set. *Principles of Data Mining and Knowledge Discovery*, pages 575–580, 2000.

[22] P. Pudil, J. Navovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994.

[23] M. Turk and A. Pentland. Face recognition using eigenfaces. *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[24] A. White. *Vegetation Variability and its Hydro-Climatologic Dependence.* PhD thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, 2005.