5A.6 ADaM-IVICS: A software tool to mine satellite data

Todd Berendes[1], Rahul Ramachandran[2*], Sara Graves[2], John Rushing[2]
[1] Earth System Science Center, Univ. of Alabama in Huntsville, Huntsville, AL.
[2] Information Technology and Systems Center, Univ. of Alabama in Huntsville, Huntsville, AL.

1. INTRODUCTION

The Algorithm Development and Mining System (ADaM) is a data mining toolkit designed for use with scientific and image data. It includes classification, clustering, feature selection, model validation, data cleaning, image processing, optimization, and association rule mining capabilities. The system consists of a set of individual algorithms or components that can be put together to perform complex tasks. Components are packaged as stand alone executables and as Python modules for easy scripting. The Interactive Visualizer and Image Classifier for Satellites (IVICS) was developed as a visualization tool to facilitate selection of training samples from satellite images for the purpose of training supervised classifiers. It has evolved into a general purpose visualization system which supports data from many satellite sensors and other scientific data sources. IVICS has been integrated with the ADaM toolkit, providing users with an end-to-end capability to interactively visualize and analyze image data while exploiting the large suite of mining algorithms available in ADaM. This paper will describe the capabilities and limitations of ADaM and IVICS. The motivation for integrating these two tools will be described along with the software engineering strategy employed for the integration to minimize modifications in either tool. Capabilities of this integrated tool will be demonstrated by stepping through two example applications.

2. ADaM

The ADaM system *(Hinke et al., 1997a, b; Ramachandran et al., 2001)* was originally developed in response to a NASA Research Announcement with the goal of mining large scientific data sets for geophysical phenomena detection and feature extraction, and has continued to be expanded and improved. Thus, unlike most data mining software, ADaM has been designed for use with scientific and image data from the outset. ADaM includes not only traditional data mining capabilities such as pattern recognition, but also image processing and optimization capabilities, and many supporting data preparation algorithms that are useful in the mining process. ADaM was recently redesigned as a toolkit of discrete, independent components to better serve the evolving service oriented computing landscape. These components can be used together in different combinations to perform many complex tasks. This redesign allows the algorithms in ADaM to be easily packaged as grid or web services *(Rushing et al., 2005)* and is being extensively used by different research groups and projects *(Droegemeier et al., 2005, Graves et. al., 2007).*

2.1 ADaM Data Mining and Pattern Recognition Capabilities

ADaM includes classification, clustering, and feature selection / reduction techniques as well as a number of utilities that are useful in pattern recognition applications. Supervised classifiers generally consist of two components: a training module and an application module. The training module uses sample patterns to learn the characteristics of the classes of interest. The application module reads the description produced by the training module and classifies patterns over a large data set. Clustering tools or unsupervised classifiers require no training step. Rather, they take a set of patterns as input and group them into classes based on similarity. The clustering tools will

Rahul Ramachandran
Information Technology and Systems Center,
University of Alabama in Huntsville,
Huntsville, AL 35899
Tel: 256-824-5157
email: rramachandran @itsc.uah.edu

produce a classified pattern set and a description of the clusters. Feature selection and reduction techniques reduce the size of the input data set by choosing a subset of the available attributes or by creating a mapping of the original feature space onto a feature space of smaller dimension. Reducing the number of features or attributes used for classification can often result in greater classification accuracy, faster classification, or both. ADaM also has data preparation utilities that aid in the pattern recognition process. Normalization is an important step that can improve the results produced during clustering and classification. The discretization utility converts numeric data into ordinal data for use in association mining or other operations that require discrete data. There are also utilities for subsetting, subsampling and cleaning the data.

2.2 ADaM Image Processing Capabilities

ADaM also provides a set of image processing modules that are useful for extracting features from images as a precursor to mining or pattern recognition. These operations typically take one or more images as input, and produce one or more images as output. They make use of ADaM's image data model (described in the next section), which supports single plane, three-dimensional images. The toolkit comes with a few translation utilities that convert to and from popular image formats such as GIF and GeoTIFF. ADaM includes basic image operations for changing the size, orientation, scale and other properties of images. It also includes level mapping utilities such as histogram equalization, inversion, thresholding and quantization. ADaM's image segmentation utilities find boundaries, contiguous regions, and polygons in images. Filtering plays an important role in many image analysis applications. ADaM has spatial domain, median, mode and morphological filters. It also has the pulse coupled neural network, which can be used for image smoothing and segmentation. The Fast Fourier Transform (FFT) is used to translate between spatial and frequency domains. Texture features are often used to classify and segment images based on local image structure; ADaM has a rich set of texture capabilities.

3. IVICS

Development of IVICS was initially driven by the requirements of the Earth Observing System (EOS) Clouds and the Earth's Radiant Energy System (CERES) *(Baum, et al., 1997)* and the Advanced Spaceborne Thermal Emission and Reflection Radiometer *(Welch, et al., 1999)* programs. A polar cloud mask was needed for both projects and it was determined that a neural network based classifier was the most practical supervised classification method *(Tovinkere et al., 1993; Berendes, et al., 1999).* Like all supervised classification techniques, neural networks require labeled training samples. In the case of satellite and other image data, detailed examination of the data is required for expert identification of visual features in the imagery. After identification of image features, a method for sample selection is needed and IVICS was originally developed for that specific purpose.

3.1 IVICS Visualization and Sample Selection Features

IVICS provides visualization options and tools designed to allow data exploration and facilitate identification of image features. Imagery can be displayed using a variety of display options including three-channel red-green-blue (RGB) color composite, indexed lookup table (LUT), and colorbar displays. The RGB display option allows the user to display any combination of channels as a three-channel composite quickly and easily. Channels may be enhanced individually or as a group using linear contrast stretch, histogram equalization, and grey scale inversion. The RGB display options are very useful for identifying image features and labeling samples.

Three different image views are available in IVICS. The main image display shows a section of the full resolution image displayed along with scrollbars if it is larger than the display window. Subsampled image display windows show an overview of the entire image and allow the user to change the image area displayed in the full resolution displays quickly and easily without using scrollbars. If the image is geospatial (i.e. satellite or gridded model) and provides latitude and longitude information the map display may be used. The map display overlays coastlines and country/state boundaries on a projected view of the image.

The LUT display option provides a convenient way to visualize and verify classifier results. Classifier results can be displayed as a color coded LUT image. RGB image data and LUT results then can be displayed simultaneously in multiple display windows while the mouse pointer is simultaneously tracked at the same coordinates in all windows. Examples of this comparison method will be shown in sections 3 and 4.

Sample selection is performed by dragging a rectangular area in an IVICS display window. The selected sample is displayed in the IVICS sample editor for further analysis and labeling. Histogram and scatter plot tools are available and the sample can be magnified using a zoom feature. After the user has labeled the sample it is added to the current sample list which may be saved as a file for use as classifier training input.

4. INTEGRATING ADaM AND IVICS

4.1 Motivation

Even though ADaM provides a large suite of algorithms, it lacks data visualization capability. Typically, ADaM users utilize their own visualization software such as IDL or Matlab to visualize the data before composing a mining workflow using ADaM modules. In some case, this flexibility is important and in other cases this dependency on additional software becomes a shortcoming. Furthermore, supervised classification requires domain experts being able to visualize the data and create samples for specific classes. Again, ADaM does not provide any general purpose tool with such capability.

On the other hand, IVICS provides functionality to visualize different imagery data and the capability to create samples for training classifiers. However, IVICS only has a limited set of operations for classification and image processing. Integrating ADaM-IVICS combines the advantages from the two systems and addresses their individual drawbacks. The integrated ADaM-IVICS system is thus a complete tool that provides end-to-end for image analysis and classification capabilities.

4.2 Integration Approach

The integration approach used was based on the motivation to minimize changes to both of the existing tools as too many changes would not justify the integration. In addition to the effort involved, it would in essence mean creating a new tool. Both the tools work on their specific data models Rather than trying to design a single model that is a union of the two existing models, a set of translation utilities were designed. The two data models and the translation utilities are described next.

4.2.1. ADaM Data Models

ADaM provides two distinct capabilities- image processing capabilities and pattern analysis capabilities; and these two types of capabilities are distinct in the types of data on which they operate. Therefore, ADaM uses two different data models: one for images and another for pattern data.

The image data model is extremely simple. An image is represented as a three dimensional array of pixel values, which are referenced by x, y and z coordinates. Two-dimensional images have z size of one. Multispectral image data can be represented using arrays of single plane images. The image data model provides methods to get and set pixel values, find the size of the image, and read and write binary image files.

Pattern vectors are represented by the ADaM pattern set class. A pattern set may have any number of attributes associated with it, and may contain an arbitrarily large number of patterns. The pattern set allows for both numeric and categorical attributes, and pattern sets may consist of mixed types of attributes. The pattern set is represented as an array of pattern vectors, with associated descriptors for each attribute. The attribute descriptors have the names of the attributes, their types, and their range of legal values. One attribute may be designated as a class attribute. The pattern set data model provides methods to add or remove attributes, add or remove pattern vectors, get and set vector values, find attribute names and properties, and read and write pattern data files. This data model is stored as Attribute-Relation File Format (ARFF) .

ADaM toolkit provides utilities to convert from image data model to pattern vector data model and vice-versa.

4.2.2. IVICS Data Model

IVICS uses the Generalized Satellite Format (GSF) which was designed specifically for IVICS. GSF was originally designed to provide a single platform and sensor independent data format for remote sensing satellite data. GSF represents satellite and other image data in a generalized data model providing a single interface for data access. GSF conversion programs have been developed for AVHRR, MODIS, ASTER, GOES, Landsat, and several other satellite data formats.

GSF files consist of an image header, channel headers, and image data. The image header stores information about satellite sensors, image dimensions, and user specified metadata. The channel headers store spectral scaling information and statistics which allow IVICS to perform on-the-fly image enhancements. Image data is stored in a band interleaved by line (BIL) format which provides a convenient and memory efficient method of access to very large images. If available, geospatial information such as latitude, longitude and solar angles may also be stored in a GSF image.

4.2.3. Coupling via Data Model Translations

The design philosophy used in ADaM is to keep the data models simple and provide a set of utilities that allow user to convert from one data model to another. The use of a simpler data model allows easier addition of new algorithms to the toolkit. The same principle of loose coupling was followed while integrating ADaM and IVICS. A set of utilities were written that convert GSF to either image or pattern vector data model and vice-versa. This approach required minimum changes to both ADaM and IVICS. The translation routines also allow the results from an ADaM mining workflow to be imported back as GSF files into IVICS for visualization. Some additional changes were made to IVICS. These changes allow the tool to save samples directly as pattern vectors for use in training supervised classifiers. These changes also allow some of the ADaM modules to be directly executed from the IVICS interface.

4. EXAMPLE APPLICATIONS

The ADaM-IVICS capabilities will be demonstrated by using the tool in two different applications. The first application focuses exploratory data analysis using unsupervised techniques such as clustering. The second application focuses on supervised classification. This application uses IVICS to visualize the data and create samples. ADaM is used to create a mining workflow. The results are then visually evaluated using IVICS.

4.1 Cluster Analysis

Cluster analysis is the name given for a group of techniques whose primary purpose is to group objects in the data based on characteristics they possess. Cluster analysis classifies objects so that each object is very similar to the other based some similarity or resemblance metric. For this example, we will try to create a cloud mask for GOES data using a clustering algorithm. The GOES data used contains five channels, one in the visible spectrum and the other four in the infrared spectrum.

Screen shots depicting the process can be seen in Fig 1 and 2. IVICS can be used to display the different channels in separate windows (Fig. 1). We can now select any of the number of clustering algorithms from the IVICS main menu (Fig 2). For this example, we select K-Means algorithm. IVICS opens a dialog box that displays the different parameters required to run the K-Means clustering algorithm (Fig. 2). We specify a channel name “K-Means” and the result from the clustering will be stored in this channel. Next, we select the different spectral channels we want to use for clustering and specify the number of clusters. For this example, we select all the five channels and set the number of clusters to two classes: cloud pixel or a non-cloud pixel. The dialog box also provides the option to select a normalization preprocessing operation before the clustering. In some cases, this preprocessing step of normalizing the data is needed to adjust for the differences within the data in order to create a common basis for clustering. Once the selections are made, we can click the “Run Program” button to begin the clustering. IVICS executes the ADaM K-Means module with the parameters specified. The module creates a new GSF file with an additional new channel “K-Means”. We can now load this new file using IVICS and visualize the cloud mask created by the K-Means clustering. The result from the clustering can be seen in Fig 3.

4.2 Supervised Classification

The real power of coupling ADaM-IVICS together can be seen while performing supervised classification. The supervised classification process is a more complicated than unsupervised and is presented in Fig 4. The process has two distinct phases, training and application, which are colored in blue and red respectively in the figure. The training phase requires the expert to visually inspect the data and select samples that represent the object of interest. Once enough representative samples have been collected, they must be then divided into two sets for training and testing. The training data is used to train the classifier. The performance of the classifier is then verified on both the training and the test data. If the classifier accuracy is within the acceptable limits for the application then the process moves to the application phase. In the application phase, the classifier is applied directly to the data. The classifier uses the information learned during the training phase to create the result – the classified data.

We will demonstrate this capability of ADaM-IVICS using a simple example. The problem is still the same as the one in the previous section, i.e., to create a cloud mask for the GOES data – but in this case we will use a Bayes classifier. Once the channels are displayed, we can start creating samples. We select representative regions in the image for clouds and label them as 1. Similarly, we select non-cloudy regions and label them 0 (See Fig. 5). Once we have enough samples, we save the samples in an ARFF file so that they can be used by ADaM modules. We now compose a mining workflow to train a classifier using the ADaM modules. The workflow can be seen in Fig 6. The first module used in the workflow splits the sample into training and test data. The training data is used to train the Bayes Classifier. Both the training and the test data are then used to evaluate the classifier. The evaluation is done using a ADaM utility called ITSC_Accuracy.

The classifier application workflow can be seen in Fig 7. The different data translation utilities are used in this workflow. The GSF-to-ARFF conversion routine is used for initial translation. The classifier is then applied to the translated data. The classification results are converted from a pattern vector to an image and then combined back to GSF using two utilities. The result from the classification can then be displayed using IVICS and is seen as the cloud mask in Fig. 8.

5. SUMMARY

The ADaM toolkit has been integrated with IVICS, providing end users with the capability to interactively visualize and analyze image data while exploiting the large suite of mining algorithms available in ADaM. The motivation for integrating the two tools and the integration principle used for coupling is presented. The paper also describes the functionality of this integrated tool with two example applications focusing on unsupervised and supervised classification techniques.

6. ACKNOWLEDGEMENTS

This work was conducted as part of the LEAD project that is funded by the National Science Foundation under the following Cooperative Agreements: ATM-0331594, ATM-0331591, ATM-0331574, ATM-0331480, ATM-0331579, ATM03-31586, ATM-0331587, and ATM-0331578.

7. REFERENCES

Baum, B. A., R. M. Welch, P. Minnis, L. L. Stowe, J. J. A. Coakley, Q. Trepte, P. Heck, X. Dong, D. Doelling, S. Sun-Mack, T. Murray, T. Berendes, K.-S. Kuo, and P. Davis, 1997: Clouds and the earth's radiant energy system (CERES) algorithm theoretical basis document subsystem 4.1 - cloud mask.

Berendes, T. A., K. S. Kuo, R. M. Welch, B. A. Baum, A. Pretre, A. M. Logar, E. C. Corwin, R. C. Weger, 1993: A comparison of paired-histogram, maximum likelihood and neural network approaches for daylight global cloud classification using AVHRR imagery. J. Geophys. Res, 104, 6199-6213.

Berendes, T. A., R. M. Welch, U. S. Nair, D. A. Berendes, 2001: Interactive visualizer and image classifier for Satellites (IVICS). Proc. AGU 2001 Spring Meeting, Boston, MA.

Droegemeier, K. K., D. Gannon, D. Reed, B. Plale, J. Alameda, T. Baltzer, K. Brewster, R. Clark, B. Domenico, S. Graves, E. Joseph, V. Morris, D. Murray, R. Ramachandran, M.

Ramamurthy, L. Ramakrishnan, J. Rushing, D. Weber, R. Wilhelmson, A. Wilson, M. Xue, and S. Yalda, 2005: Service-Oriented Environments in Research and Education for Dynamically Interacting with Mesoscale Weather. IEEE Computing in Science & Engineering, 7, 24-32.

Graves, S., R. Ramachandran, K. Keiser, M. Maskey, C. Lynnes, and L. Pham, 2007: Deployable Suite of Data Mining Web Services for Online Science Data Repositories. 87th AMS Annual Meeting, San Antonio, TX.

Hinke, T., J. Rushing, S. Kansal, S. J. Graves, H. S. Ranganath, and E. Criswell, 1997a: Eureka Phenomena Discovery and Phenomena Mining System. AMS 13th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography and Hydrology.

Hinke, T., J. Rushing, H. S. Ranganath, and S. J. Graves, 1997b: Target-Independent Mining for Scientific Data: Capturing Transients and Trends for Phenomena Mining. Proceedings Third International Conference on Data Mining (KDD-97), Newport Beach, California.

Ramachandran, R., H. Conover, S. J. Graves, K. Keiser, S. Movva, and S. Tanner, 2001: Flexible Earth Science Data Mining Architecture. Fourth Workshop on Mining Scientific Datasets, Seventh ACM SigKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA.

Rushing, J., R. Ramachandran, U. Nair, S. Graves, R. Welch, and A. Lin, 2005: ADaM: A Data Mining Toolkit for Scientists and Engineers. Computers & Geosciences, 31, 607-618.

Tovinkere, V. R., M. Penaloza, A. Logar, J. Lee, R. C. Weger, T. A. Berendes, R. M. Welch, 1993: An Intercomparison of Artificial Intelligence Approaches for Polar Scene Identification. Journal of Geophysical Research,, 98, 5001 - 5016.

Welch, R. M., D. Berendes, T. Berendes, K.-S. Kuo, A. M. Logar, 1999: The ASTER Polar Cloud Mask Algorithm Theoretical Basis Document.

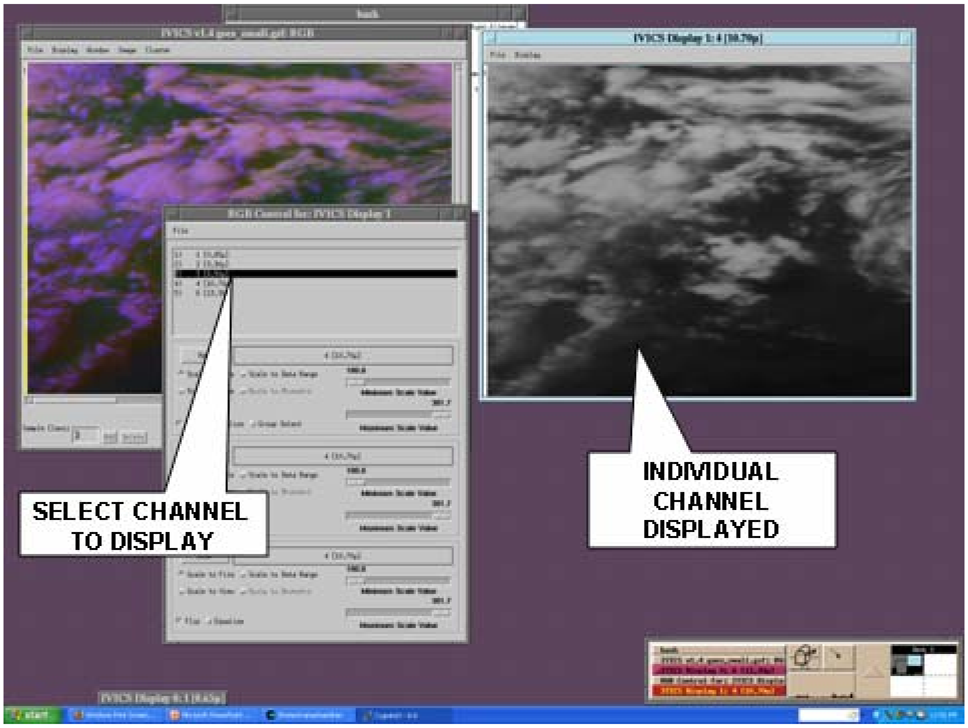

IVICS v1.4 goes_small.gif RGB
IVICS Display 1: 4 [10.70µ]
RGB Control for: IVICS Display 1
SELECT CHANNEL TO DISPLAY
INDIVIDUAL CHANNEL DISPLAYED
start

Fig 1: Visualizing GOES data using IVICS

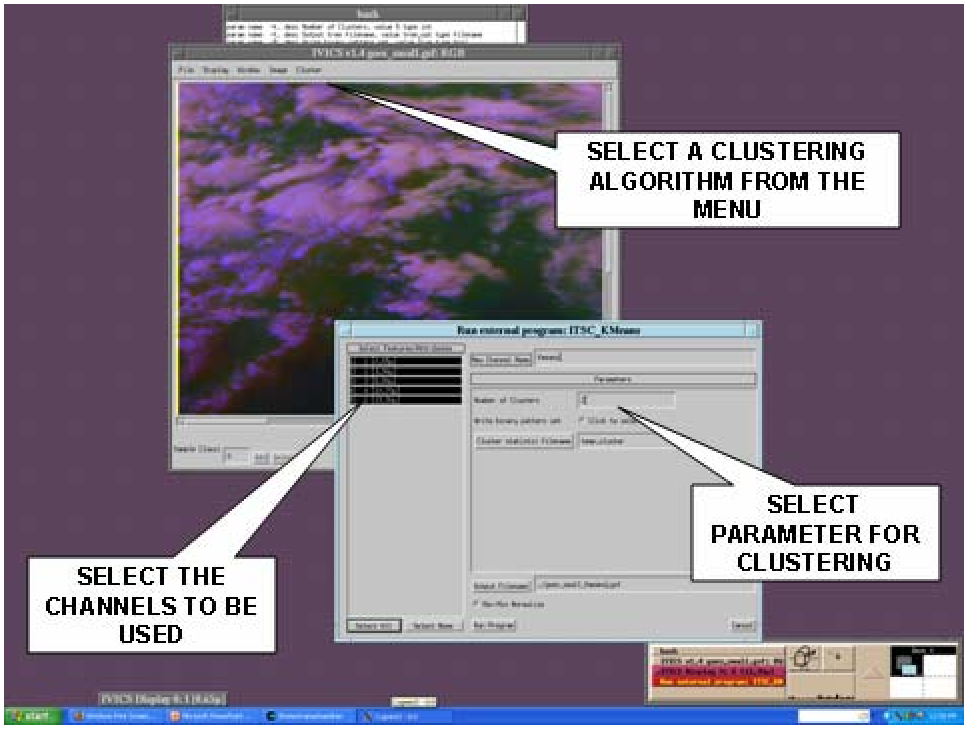

IVICS v1.4 goes_small.gif: RGB
SELECT A CLUSTERING ALGORITHM FROM THE MENU
Run external program: ITSC_KMeans
SELECT PARAMETER FOR CLUSTERING
SELECT THE CHANNELS TO BE USED
IVICS Display 0: 1 [0.63µ]

Fig 2: Selecting the clustering algorithm and setting its parameters in IVICS

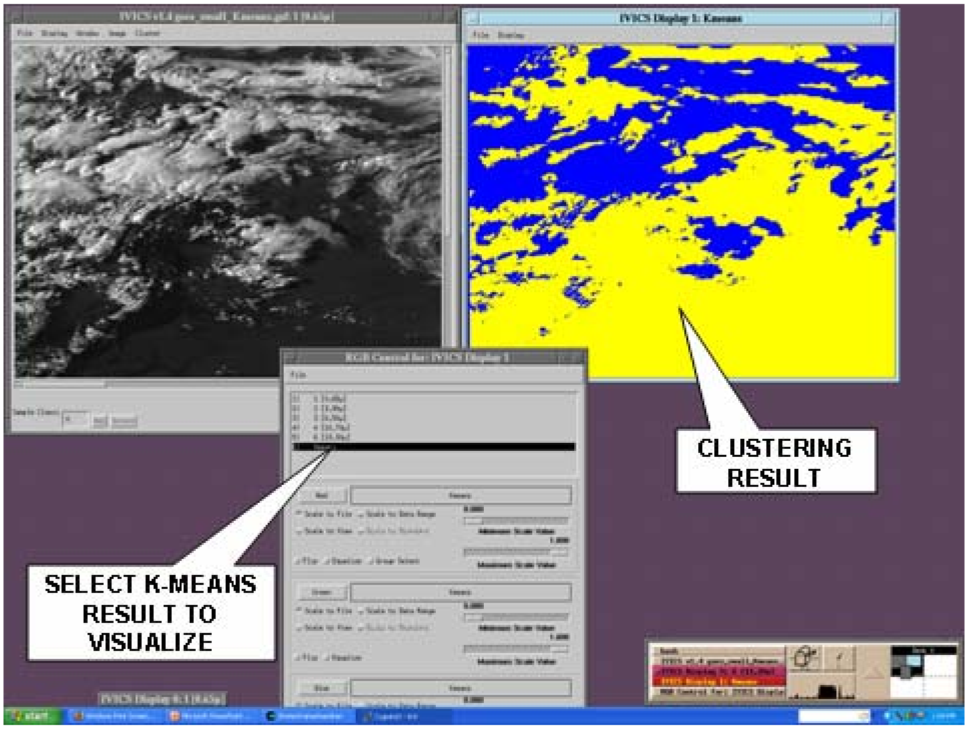

IVICS v1.4 goes_small_Kmeans.gif: 1 [0.63µ]
IVICS Display 1: Kmeans
RGB Control for: IVICS Display 1
CLUSTERING RESULT
SELECT K-MEANS RESULT TO VISUALIZE
IVICS Display 0: 1 [0.63µ]

Fig. 3: Visualizing the clustering results

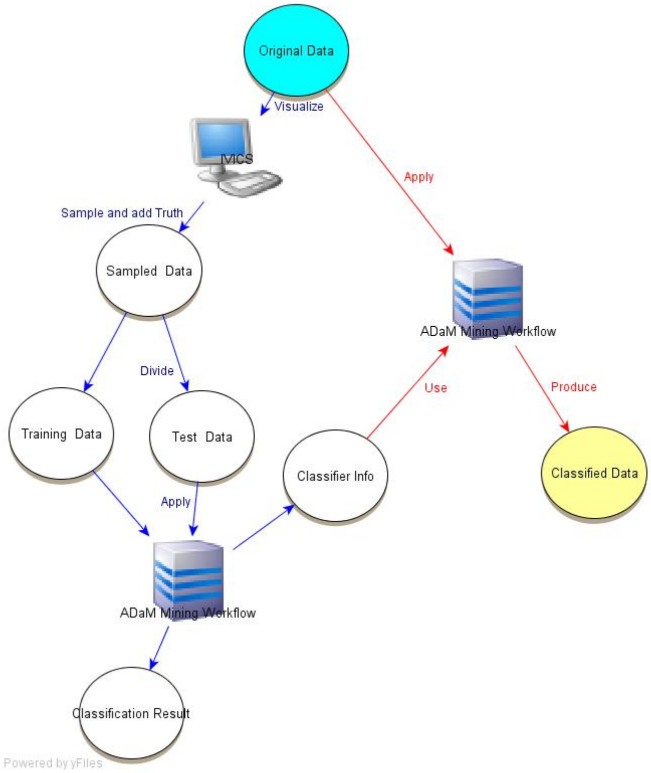


Fig. 4: Supervised classification process

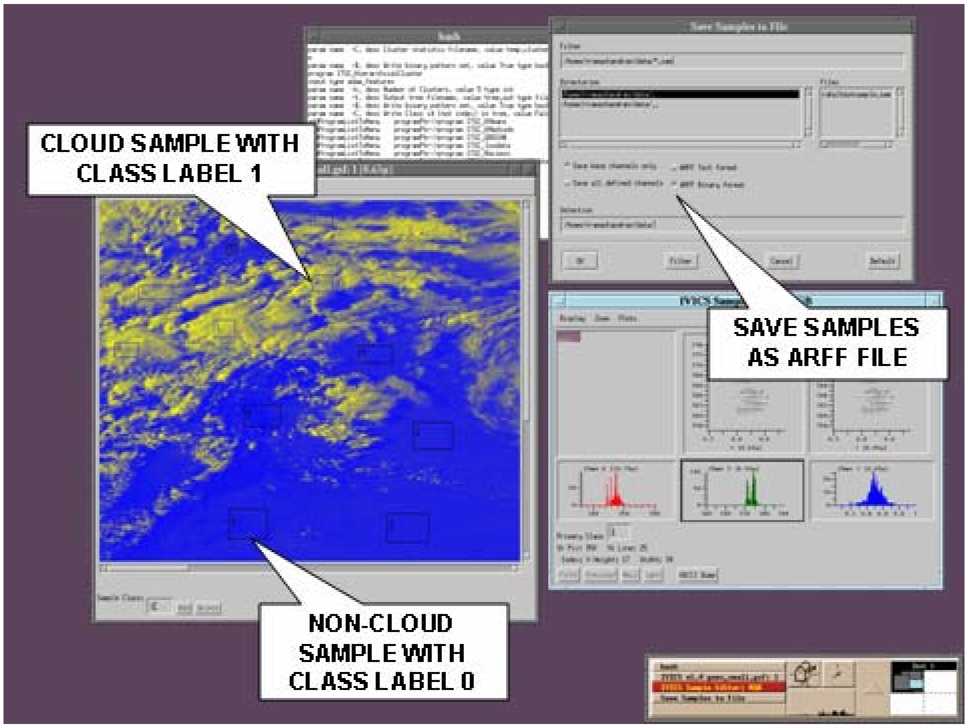

CLOUD SAMPLE WITH CLASS LABEL 1
SAVE SAMPLES AS ARFF FILE
NON-CLOUD SAMPLE WITH CLASS LABEL 0
Save Samples to File
bash
IVICS Sample Editor

Fig. 5: Sample selection using IVICS

```
# Split the sample data into training and testing files
ITSC_sample -c class -i goesSamples.arff -o goesTrain.arff -t goesTest.arff -p 0.50 –B

# Train and Test the classifier
ITSC_BayesclassifierTrain -b bayes.txt -c class -i goesTrain.arff

# Run the classifier on the training data
ITSC_BayesclassifierApply -b bayes.txt -c class -i goesTrain.arff -o goesTrainResult.arff –B

# Run the classifier on the test data
ITSC_BayesclassifierApply -b bayes.txt -c class -i goesTest.arff -o goesTestResult.arff –B

# Evaluate the results for:
# 1. Training Data
ITSC_Accuracy -c class -t goesTrainResult.arff -v goesTrain.arff
# 2. Test data
ITSC_Accuracy -c class -t goesTestResult.arff -v goesTest.arff
```

Fig. 6: Mining workflow to train a Bayes classifier

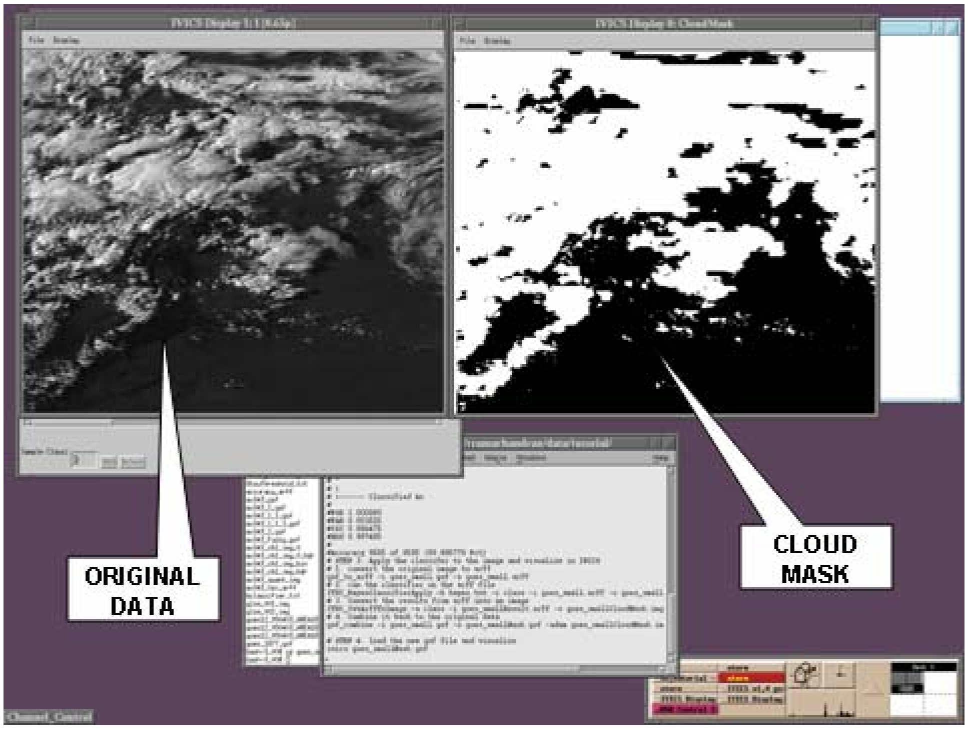
```
# Apply the classifer to the image and visualize in IVICS
# 1. convert the original image to arff
gsf_to_arff -i goes_small.gsf -o goes_small.arff

# 2. run the classifier on the arff file
ITSC_BayesClassifierApply -b bayes.txt -c class -i goes_small.arff -o goes_smallResult.arff -B

# 3. Convert the results from arff into an image
ITSC_CvtArffToImage -a class -i goes_smallResult.arff -o goes_smallCloudMask.img

# 4. Combine it back to the original data
gsf_combine -i goes_small.gsf -o goes_smallMask.gsf -adam goes_smallCloudMask.img -adam_label
'CloudMask'
```

Fig. 7: Mining workflow to apply the trained classifier on GOES data



```
#
# ------- Classified As
#
[illegible]
#
# STEP 3. Apply the classifier to the image and visualize in IVICS
# 1. convert the original image to mff
gof_to_mff -i goes_small.gof -o goes_small.mff
# 2. run the classifier on the mff file
[illegible] -i goes_small.mff -o goes_small
# 3. Convert the results from mff into an image
[illegible]
# 4. Combine it back to the original data
gof_combine -i goes_small.gof -o goes_smallMask.gof [illegible]

# STEP 4. load the new gof file and visualize
ivics goes_smallMask.gof
```

Fig. 8: Cloud mask created by the Bayes classifier