Mining High-Dimensional Data for Information Fusion: A Database-Centric Approach

Boriana L. Milenova
Data Mining Technologies
Oracle
Boriana.Milenova@oracle.com

Marcos M. Campos
Data Mining Technologies
Oracle
Marcos.M.Campos@oracle.com

Abstract - *Data mining on high-dimensional heterogeneous data is a crucial component in information fusion application domains such as remote sensing, surveillance, and homeland security. The information processing requirements of these domains place a premium on security, robustness, performance, and sophisticated analytic methods. This paper introduces a database-centric approach that enables data mining and analysis of data that typically interest the information fusion community. The approach benefits from the inherent security, reliability, and scalability found in contemporary RDBMSs. The capabilities of this approach are demonstrated on satellite imagery. Hyperspectral data are mined using clustering (O-Cluster) and classification (Support Vector Machines) techniques. The data mining is performed inside the database, which ensures maintenance of data integrity and security throughout the analytic effort. Within the database, the clustering and classification results can be further combined with spatial processing components to enable additional analysis.*

Keywords: Information fusion, high-dimensional data, database, support vector machines, clustering.

1 Introduction

Information fusion has become an area of growing importance as rapid advances in sensor technology have produced a variety of powerful and low-cost sensor arrays. Such sensors can be used to enable information fusion applications in many domains, including: remote sensing, surveillance, homeland security, and health management. With the continuing expansion of the domain of interest and the increasing complexity of the collected information, data mining on high-dimensional heterogeneous data has become a crucial component in information fusion applications [1, 2, 3].

The information processing requirements of many fusion application domains place emphasis on security, reliability, scalability, and sophisticated analytic methods. Due to the specialized nature of these tasks, many endeavors result in ad-hoc prototypes that fail to completely satisfy the domain requirements. Integration and re-use of available technology is also problematic. This situation is compounded by the lack of common engineering standards for data fusion systems [4]. A platform providing computational components in an integrated framework, along with established standards, would clearly be advantageous in achieving fast delivery of mature, robust applications. Faster prototyping would allow more reviews of product deliverables by customers and would result in better overall product quality.

This paper proposes that modern RDBMSs, with their capabilities for supporting mission critical applications, distributed processing, security, and integration of analytics, are an ideal platform for the implementation of information fusion applications. Given the data-centric nature of the information fusion process, leveraging existing RDBMS infrastructure allows for an efficient and effective information fusion system implementation. In this proposal, data mining plays a key role given its increasing importance in information fusion systems. Waltz [1] outlines the complementary roles of data mining and data fusion – the data mining driven process of automatic model discovery can be integrated into the data fusion driven automatic target identification activity.

The paper is organized as follows. Section 2 outlines the proposed database-centric approach and highlights some of the key components. Section 3 provides an illustration of the capabilities of the system for analyzing hyperspectral satellite imagery. The approach is demonstrated with references to existing functionality in the Oracle Database 10g Release 2.

2 Analytic data warehousing

An Analytic Data Warehouse (ADW) is a data warehouse with analytic capabilities. Incorporating analytics into a data warehouse provides many advantages over performing data analysis outside the data warehouse. Data integrity, security, and performance scalability are inherent in modern RDBMSs. Such requirements are also relevant to information fusion applications; commonly constructed as standalone applications, they incur costly (and often prolonged) creation of infrastructure necessary to achieve security and scalability.

The importance of a tight coupling between data integration and analysis for a successful information system and the advantages of database techniques in achieving this requirement were emphasized recently in the InFuse system [5]. Unlike ADW, the InFuse fusion

engine is located outside the RDBMS so it suffers from the drawbacks common to such systems as outlined in the preceding paragraph.

An information fusion ADW-based architecture (Figure 1) typically includes the following major components:

- Sensor arrays
- Extraction, transformation and load (ETL) of sensor data
- Centralized data warehousing
- Analytic module
- Visualization and reports

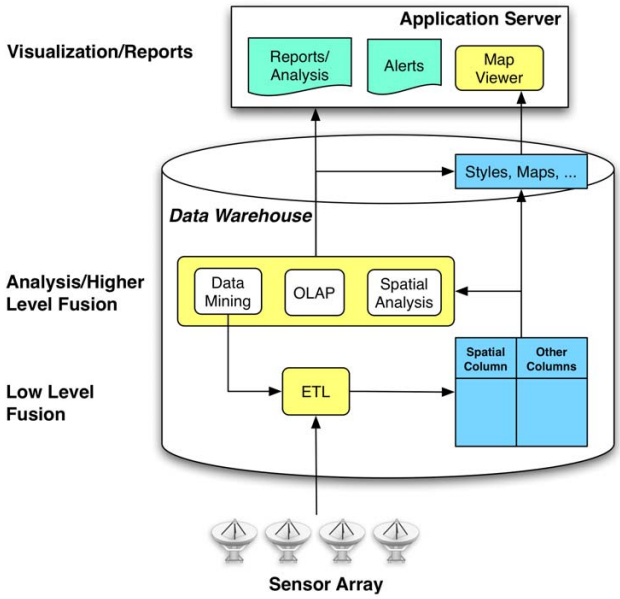


Figure 1. ADW-based information fusion architecture.

Sensor arrays produce streams of data that need to be fused and analyzed. Sensor data are processed and loaded into a centralized data repository. The generation of spatial features can be also part of ETL. All the required analytical methods are integral to the database infrastructure – no data movement is required. The stored data can be used for data mining model generation. The generated models can undergo scheduled distribution and deployment across different database instances. These models can then be used for monitoring the incoming sensor data. The database can issue alerts when suspicious activity is detected. The models and the stored sensor data can be further investigated using database reporting and analysis tools (e.g., via OLAP).

The key aspect to the described data flow is that processing is entirely contained within the database. With the exception of the sensor array, all other components can be found in modern RDBMSs. Among the major benefits of using such an integrated approach are improved security, speed, data management and access, and ease of application development. In comparison with stand-alone applications, significant computational savings can be derived from eliminating the need for data export out of the database and leveraging database parallelism. The amount of savings is application specific.

The individual components in ADW relate to the revised Joint Directors of Laboratories (JDL) Data Fusion Process model [4, 6, 7] as follows: Sensor Array (Sources of Information), ETL (Levels 0 and 1 processing), Analytic Module (Levels 1, 2, and 4 processing), Reports, Visualization, and Alerts (Level 5 processing). The database-centric approach facilitates monitoring and analysis of user-system interactions. This can be leveraged for effectively supporting Level 5 activities. The following sections discuss the functionality and usage of the individual components.

2.1 Sensor array

A wide variety of sensor types exists. Sensor arrays can contain a single type or multiple types of sensors (e.g., radar, infrared, hyperspectral). Sensors can be either local or distributed, using remote connections to the fusion system. Data from sensor arrays can be also complemented with other data, such as geographical information [6].

2.2 ETL

Sensor streams may require further pre-processing and feature extraction before the data can be successfully used for data mining and analysis. This stage performs low-level data fusion. For example, sensor data can be transformed into a consistent set of units and coordinates [6]. In the RDBMS context, SQL and user-defined functions offer a great degree of flexibility and efficiency in extracting key pieces of information from the data stream. Useful SQL capabilities include mathematical, aggregate, and analytic functions. For example, windowing functions can be used to compute aggregates over time intervals or number of rows.

2.3 Data warehouse

Using an RDBMS as a centralized data repository offers great flexibility in terms of data manipulation. Inputs from different sources can be easily combined through joins. Without replicating data, database views can capture different slices of the data (e.g., data over a given time interval, data for a specific sensor). Such views can be then used directly for model generation and data analysis. This information can also be used to facilitate data fusion process refinement. An RDBMS has the additional benefits of data security, high availability, high load support, and fast response time.

2.4 Analytic module

This module carries out analysis of the data and information fusion that can be implemented with machine

learning/data mining algorithms. It can also perform OLAP and spatial analysis. The output of the analysis can be captured in reports and maps that are displayed in the reporting module. The data mining component combines automated model generation and distribution, as well as real-time and offline detection.

Model generation

Data mining techniques that have been used in the context of information fusion include maximum likelihood classifiers, neural networks, decision trees, and support vector machines (SVM) [3, 8]. Modern RDBMSs offer, to different degrees, robust and effective implementations of data mining techniques that are fully integrated with core database functionality. The incorporation of data mining eliminates the necessity of data export outside the database, thus enhancing data security. Since the model representation is native to the database, no special treatment is required to ensure interoperability.

In order to programmatically operationalize the model generation process, data mining capabilities can be accessed via APIs (e.g., JDM standard Java API, PL/SQL data mining API). Specialized GUIs as entry points can be also easily developed, building upon the available API infrastructure (e.g., Oracle Data Miner). Such GUI tools enable interactive data exploration and initial model investigation.

Model distribution

In ADW, model distribution is greatly simplified since models are not only stored in the database but are also executed in it as well. Models can be periodically updated by scheduling automatic builds. The newly generated models can then be automatically deployed to multiple database instances.

Information fusion applications implemented within a modern RDBMS framework can transparently leverage the grid computing infrastructure available for the database. Grids make possible pooling of available servers, storage, and networks into a flexible on-demand computing resource capable of achieving scalability and high availability [9, 10]. An example of a grid computing infrastructure is Oracle's Real Application Clusters (RAC) architecture. RAC allows a single Oracle database to be accessed by concurrent database instances running across a group of independent servers (nodes).

A grid-enabled RDBMS system needs to be seamlessly integrated with a scheduling infrastructure. Such an infrastructure enables scheduling, management, and monitoring of model build and deployment jobs. An example of a scheduling system meeting the above requirements is Oracle Scheduler.

Model scoring and alerts

Detection (e.g., target identification) can be performed either real-time or offline. Real-time detection (and alarm generation) is essential for the instrumentation of many information fusion applications.

In the context of ADW, an effective real-time identification mechanism can be implemented by leveraging the parallelism and scalability of a modern RDBMS. This removes the need for a system developer to design and implement such infrastructure. Inside the database, detection can be tightly integrated, through SQL, into the ETL process itself. This is indicated by the arrow connecting data mining and ETL in Figure 1.

Applications must be able to generate alarms, notify interested parties, and possibly initiate responses. Such requirements can be easily satisfied in ADW using existing RDBMS infrastructure. Database triggers are powerful SQL mechanisms that initiate predefined actions when a specific condition is met. Oracle's Publish-Subscribe messaging infrastructure can support asynchronous communications in distributed systems that operate in a loosely-coupled and autonomous fashion and require operational immunity from network failures.

2.5 Visualization and reports

Using a database as the platform for information fusion applications facilitates the generation of data analysis results and reports. Collected sensor data, target identification predictions, as well as model contents, can be inspected either directly using queries or via higher level reporting and visualization tools (e.g., Discoverer, Oracle Reports). Analysis results can also be leveraged by a large set of report and interface development tools. For example, web portal creation tools (e.g., Oracle Portal) offer infrastructure for the development of application 'dashboards'. This allows circumvention of a lengthy application development process and provides standardized and easily customized report generation and delivery mechanisms (e.g., Oracle MapViewer and Application Server).

3 Examples on hyperspectral data

The capabilities of ADW for supporting information fusion (Level 0-1) are illustrated on a hyperspectral satellite image. Emphasis is placed on the data mining and spatial analysis modules and their integration. The spectral bands are stored in a database table (one record per pixel). The individual bands can either be stored in distinct table columns or in a single nested table column. A spatial column of type `SDO_GEOMETRY` encodes the coordinate information for each pixel. Data mining and spatial analysis can be performed directly on this table within the RDBMS. The following sections demonstrate the creation of annotated maps that produce a faithful representation of the underlying features in the image. Unsupervised (clustering) and supervised (classification) mining approaches are described. The data mining results are further combined with Oracle spatial functionality for exploring areas of interest in the map.

Methodological approaches and result quality are discussed throughout the exposition. However, the present study does not attempt to achieve the best performance on the selected dataset via customized pre-processing, parameter tuning, committee of models, or other types of optimization. Instead, the focus is on how existing ADW functionality can be leveraged for rapid prototyping in data analysis and modeling with the ultimate goal of fast delivery of reliable applications.

3.1 Hyperspectral dataset

The hyperspectral image data were acquired from the AVIRIS sensor. This image has been used in many previous studies [8, 11, 12, 13] and is publicly available (ftp://ftp.enc.purdue.edu/biehl/MultiSpec/92AV3C). The data was converted into ASCII format using the MultiSpec image processing tool [14]. The image was captured over a rural area in the northern part of Indiana (Figure 2a). Most of the scene represents agricultural land and wooded areas. The image also includes some buildings and road infrastructure. A field survey map is available and has been used as a ground truth reference in previous experimental studies. The survey information is high-level (polygon rather than pixel based) and incomplete – it covers only about 50% of the image and omits some prominent features. The current set of experiments uses this reference information for evaluation of the unsupervised method results and as a source of labels for the supervised classification method.

The AVIRIS image has 224 bands and consists of 145x145 pixels. Four of the bands do not contain data. Following the procedure from previous studies on the same dataset [13], 20 water absorption bands (104-108, 150-163, and 220) and 15 noisy bands (1-3, 109-112, 148, 149, 164, 165, and 217-219) were removed from the data, thereby reducing the number of bands to 185.

Typically, the hyperspectral data can be subjected to a number of pre-processing steps (e.g., contrast enhancement, texture processing [15, 16]). In addition, to avoid the "curse of dimensionality" associated with mining high-dimensional spaces, dimensionality reduction techniques are often employed [17, 18]. Data transformations and preprocessing can be easily integrated within an RDBMS framework (e.g., Oracle's PL/SQL procedures, table functions). While raw image data pre-processing is an important research area, in the present work information fusion is accomplished at a later stage of the processing flow – in the analytic module. Here, we use the sensor data from the selected 185 hyperspectral bands directly and employ data mining algorithms well suited for high-dimensional spaces.

3.2 Clustering

Clustering is an unsupervised data mining technique that can be used to derive new concepts from an array of low level features. Many popular clustering algorithms (e.g., k-Means [19], DBSCAN [20]) rely on near or nearest neighbor information. Such distance-based methods are not fully effective in high-dimensional spaces. As dimensionality increases, the data space becomes sparsely populated and the distances between individual data points become very similar. That is, the difference between the distance to the nearest and farthest neighbors of a data object may approach zero [21].

Density-based clustering algorithms (e.g., CLIQUE [22], OptiGrid [23]) have been successfully employed to address the nearest neighbor problem in high-dimensional spaces. These methods typically divide the input space into hyper-rectangular cells. Cells with high data density are interpreted as clusters.

Oracle's orthogonal partitioning clustering (O-Cluster) is a density-based method that was developed to handle large high-dimensional databases [24, 25]. O-Cluster uses a top-down partitioning strategy based on orthogonal projections of the data to identify areas of high density. The algorithm attempts to separate dense contiguous regions into individual clusters. O-Cluster computes uni-dimensional histograms along individual input attributes.

a) b)

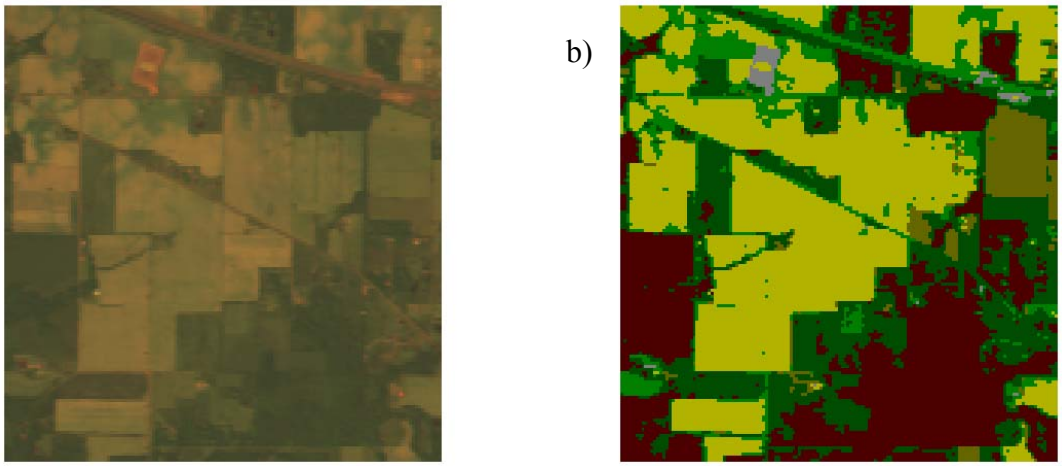


Figure 2. Hyperspectral image: a) original data using bands 35, 130, and 194; b) O-Cluster results.

The algorithm searches for splitting points along these histograms that would produce cleanly separable, and preferably balanced, clusters. O-Cluster operates recursively to create a binary tree hierarchy based on univariate splitting points in each inner node. The topology of the hierarchy, along with its splitting predicates, can be used to gain insights into the clustering solution. The number of leaf clusters is determined automatically.

Figure 2b shows an annotated map based on the results produced by O-Cluster. The algorithm identified six distinct clusters. Each of these clusters is assigned a different color in the map. The individual clusters capture distinctive features in the image. Further insight into the results and their quality can be gained by exploring the model. Figure 3 depicts the hierarchical clustering tree.

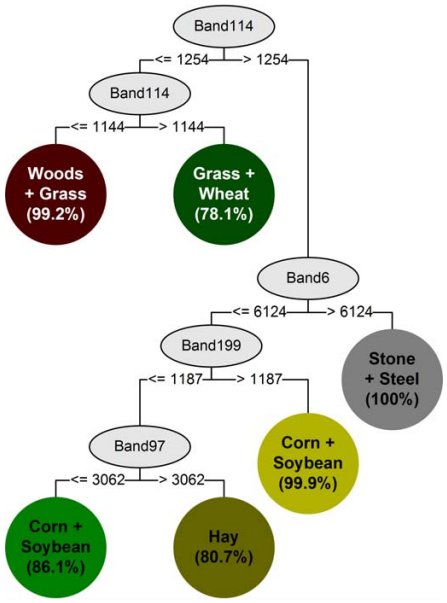


Figure 3. O-Cluster model.

Every branching node contains split predicate information (band number and the split condition). The selected bands were found to be most discriminative in the identification of dense regions with low overlap. O-Cluster's model transparency can be used to gain insight into the underlying structure of the data and can assist feature selection. The six leaf nodes in Figure 3. O-Cluster model. map to corresponding areas in Figure 2b (the same color coding is used in both figures). To assess the purity of the clusters and the quality of the results, each leaf node was labeled using ground truth information. Only pixels that were part of the survey map (~50% of the image) were used. The percentage value within each leaf indicates the fraction of survey map pixels within a given cluster that belong to the chosen label. Even though the survey information is incomplete and does not fully reflect all relevant features, the resulting clusters have reasonably high purity. The transparent nature of the hierarchy also allows the extraction of simple descriptive rules, such as:

```
If Band114 > 1254 and Band6 > 6124 then the
pixel belongs to a stone/steel structure.
```

The maps derived from the clustering model are created in a fully unsupervised manner. Such high-level maps can identify the most salient image features. They represent a good starting point that can be further improved upon by human experts or supervised data mining algorithms.

3.3 Classification

Land cover classification on the basis of multispectral data is an active research area. A number of algorithms have shown promise in producing annotated maps with high accuracy [2, 26]. Many studies focus on sensor data of low to medium dimensionality. Hyperspectral data, on the other hand, with its high dimensionality can be problematic for some classifiers popular within the domain. Shah et al. [8] investigated the accuracy of several supervised fusion algorithms on hyperspectral data and concluded that support vector machines (SVM) have superior performance compared to the maximum likelihood classifier and the back-propagation neural network fusion approaches. Another comparative study on hyperspectral data [3] produced similar findings – SVM outperformed maximum likelihood, decision trees, and neural networks.

Such results are not surprising since SVM has emerged as the data mining algorithm of choice for high-dimensional input spaces where traditional statistical modeling techniques underperform. Examples of such domains include text mining, bioinformatics, and intrusion detection. SVM's strong regularization properties ensure good generalization to novel data irrespective of the dimensionality of the input space. The algorithm seeks a decision boundary that would ensure the most robust discriminant solution, thus preventing overfitting to the training data. The decision boundary represents a hyperplane in a high-dimensional space. In the case of low-dimensional data, non-linear kernel functions can be used to transform the input space into a high-dimensional feature space. The high-dimensional feature space then gives SVM models the flexibility to fit arbitrarily complex non-linear decision surfaces accurately. For high-dimensional data (e.g., hyperspectral imagery), a non-linear kernel transformation to a higher dimensional feature space is usually not necessary.

In the present study, a linear SVM model was built. The multiclass strategy was one-vs-all (i.e., one binary model per target class). Target class information was derived from the field survey map. Eight distinct target classes were considered. Some of the crop targets represent a summary of several survey map labels – for example, there were three subtypes of corn fields (corn, corn-notill, and corn-min). Two thirds of the pixels available in these categories were used for training and one third for testing. Table 1 shows the confusion matrix on the test dataset. The rows represent actual ground truth

values while the columns represent the predictions made by the model. The overall accuracy is 85.7%. The main source of error is confusion between corn and soybean fields. There are also some misclassifications between the woods and building/grass/tree/drives categories. Another study on the same dataset [13] has shown that such errors can be significantly reduced by using a one-vs-one multiclass strategy (i.e., pair-wise models for every target class combination). Such a strategy is more costly both in terms of build and scoring times and will not be explored here.

Table 1. SVM confusion matrix; target classes: corn (C), grass (G), hay (H), soybeans (S), wheat (W), woods (D), buildings/grass/tree/ drives (B), stone/steel (SS).

	C	G	H	S	W	D	B	SS
C	538	2	0	260	0	0	0	0
G	10	393	7	7	0	2	0	0
H	0	0	170	0	0	0	0	0
S	96	8	0	1237	0	0	0	1
W	0	0	0	0	61	0	0	0
D	2	12	0	0	0	414	4	0
B	0	23	0	2	3	39	52	0
SS	2	0	0	1	0	0	0	26

Figure 4a shows a map based on SVM's predictions. While the overall map quality is reasonable and the different types of crops are easily identifiable, the noise due to the corn-soybean and woods-buildings/grass/ tree/drives errors is evident. Various approaches can smooth the predictions based on neighborhood information – one possibility would be to provide the classifier with spatial input (e.g., coordinate values or predictor averages). Here we chose to post-process the classifier predictions using Oracle spatial functionality.

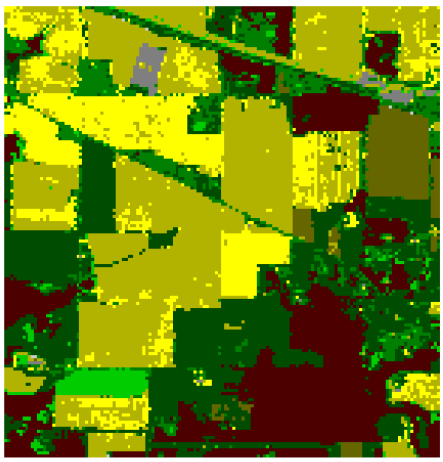
The predictions for each class can be treated as thematic layers – each layer represents a 2D array of binary values where each bit corresponds to a pixel in the image. If a bit is set, a positive prediction within this thematic layer will be made. A simple smoothing strategy would be to set or unset a bit based on the values of its immediate neighbors. Here we consider the values of 9 pixels (the center pixel and its 8 immediate neighbors). A bit is set only if the SVM classifier made positive predictions for 2/3 of the neighborhood. Otherwise the bit remains unset.

Figure 4b illustrates the results of such smoothing. The amount of noise is greatly reduced and the dominating features of the scene are well delineated – the 'smoothed' predictions have 90% accuracy. However, due to the stringent 2/3 smoothing criterion some of the transition areas result in no predictions and some localized features (e.g., correct predictions on small man-made structures) are smoothed out. The level of detail is usually application specific and can be adjusted accordingly. The smoothed results were computed using spatial queries. A spatial index was built on the table column containing coordinate data. This index can improve the performance of nearest-neighbor queries. The *k*-nearest neighbors of a point can be retrieved using the `SDO_NN` operator. Alternatively, `SDO_WITHIN_DISTANCE` retrieves all points within a given radius. The second approach is preferable here as it handles the edge effects.

To further illustrate the flexibility and expressive power of such an integrated approach, we include a sample SQL query that combines data mining and spatial features to perform the following task: Within the left upper quadrant of the image, find the soybean fields that are no further than 150m from stone-steel structures.

The optional `WITH` clause is used here for clarity and improved readability of the subqueries (lines 2-35). The first subquery, named `quadrant` (lines 2-6), restricts the

a)



b)

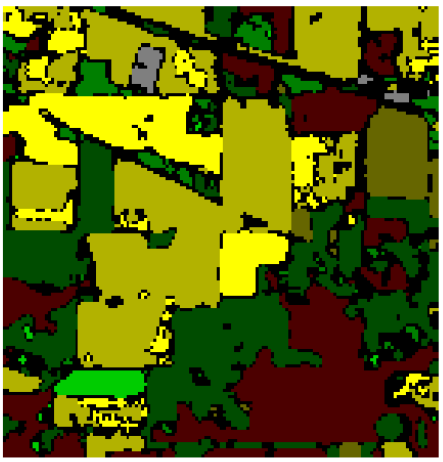


Figure 4. SVM predictions: a) raw classifier output; b) smoothed classifier output; colors: corn (bright yellow), soybeans (medium yellow), hay (dark yellow), wheat (bright green), buildings/grass/trees/drives (medium green), grass (dark green), woods (brown), stone-steel (grey); in panel b) pixels without prediction are marked in black.

```
1   WITH
2    quadrant AS(
3     SELECT *
4     FROM hyperspectral_data a
5     WHERE a.pixel.sdo_point.x < 1460
6     AND a.pixel.sdo_point.y < 1460),
7    soybean AS (
8     SELECT a.pixel.sdo_point.x x,
9            a.pixel.sdo_point.y y
10    FROM quadrant a
11    WHERE PREDICTION(
12           SVM_model
13           using *) = 'soybean'
14    AND PREDICTION_PROBABILITY(
15           SVM_model,
16           'soybean'
17           using *) > 0.5),
18   stone_steel AS (
19    SELECT pixel FROM quadrant
20    WHERE PREDICTION(
21           SVM_model
22           using *) = 'stone-steel'
23    AND PREDICTION_PROBABILITY(
24           SVM_model,
25           'stone-steel'
26           using *) > 0.5),
27   stone_steel_150_radius AS(
28    SELECT DISTINCT
29           a.pixel.sdo_point.x x,
30           a.pixel.sdo_point.y y
31    FROM quadrant a, stone_steel b
32    WHERE
33        SDO_WITHIN_DISTANCE(
34          a.pixel, b.pixel,
35         'distance=150')='TRUE')
36  SELECT a.x, a.y
37  FROM soybean a,
38       stone_steel_150_radius b
39  WHERE a.x=b.x AND a.y=b.y;
```

search to the left upper quadrant. The spatial coordinates in the left upper quadrant are smaller than 1460 (73 pixels x 20m raster grid). All subsequent subqueries run against this restricted set of points. The second subquery, `soybean` (lines 7-17), identifies the coordinates of the pixels that were classified by SVM as soybean with probability greater than 0.5. The query uses the `PREDICTION` and `PREDICTION_PROBABILITY` SQL operators. Predictions are made using the SVM model described earlier. The third subquery, `stone_steel` (lines 18-26), selects the pixels that were classified as stone-steel structures with probability greater than 0.5. Unlike the previous subquery that retrieved the pixel coordinates, here we return pixel spatial objects. These objects will be used in the final subquery, `stone_steel_radius_150` (lines 27-35). This subquery retrieves the coordinates of all pixels that fall within 150m of a stone-steel structure. It makes use of the `SDO_WITHIN_DISTANCE` operator. This operator leverages the spatial index on the pixel column to efficiently retrieve all objects within the specified radius. Since the operation is equivalent to pair-wise comparisons between the two groups of pixels, the `DISTINCT` clause limits the output to unique coordinate pairs. The main query (lines 36-39) returns the coordinates of the pixels from the upper left quadrant that are soybean fields and lie within the 150m zone. Figure 5 illustrates the results of this query.

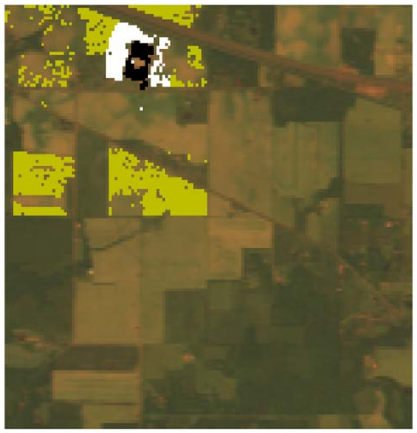


Figure 5. Query results; colors: soybean (yellow), stone-steel (black), soybean within 150m of stone-steel (white).

This example highlights the ease of combining individual ADW components and the wealth of expression that can be achieved via SQL queries. The terseness of the SQL code and its modularity are important assets in the development of complex and mission critical applications. Even though the example here performs a batch operation, the SQL operators are very well suited for real-time applications and can be integrated within the ETL process.

4 Conclusions

A database-centric platform for building information fusion applications offers many advantages. These include tight integration of individual components, security, scalability, and high availability. Current trends in RDBMSs are moving towards providing all key components for delivering comprehensive state-of-the-art information fusion applications. The Oracle Database 10g Release 2 already incorporates these key functionalities, including strong data mining and spatial analysis features. As illustrated above for a hyperspectral satellite image information fusion problem, these features provide great flexibility and analytic power. By leveraging an existing RDBMS-based technology stack, a full-fledged information fusion application can be developed in a reasonably short time and at low development cost.

References

[1] E. L. Waltz, *Information understanding: Integrating data fusion and data mining processes,* IEEE Intl. Symp. Circuits and Systems, Monterrey, CA, May 1997, Vol 6, pp. 553-556.

[2] J. T. Morgan, A. Henneguelle, J. Ham, J. Ghosh, and M. M. Crawford, *Adaptive feature spaces for land cover classification with limited ground truth data*, Intl. J.

Pattern Recognition and Artificial Intelligence, Vol 18, No. 5, pp. 777-800, 2004.

[3] M. Pal and P. M. Mather, *Assessment of the effectiveness of support vector machines for hyperspectral data*, Future Generation Computer Systems, Vol 20, No. 7, pp. 1215-1225, 2004.

[4] A. N. Steinberg, C. L. Bowman, and F. E. White, *Revision to the JDL data fusion model,* Third NATO/IRIS Conf., Quebec City, Canada, 1998.

[5] O. Dunemann, I. Geist, R. Jesse, K.-U. Sattler, and A. Stephanik, *A database-supported workbench for information fusion: InFuse*, 8th Intl. Conf. Extending Database Technology, Prague, Czech Republic, March 2002, pp. 756-758.

[6] J. Llinas and D. L. Hall, *An Introduction to Multi-Sensor Data Fusion*, IEEE Intl. Symp. Circuits and Systems, Monterrey, CA, May 1998, pp. 537-540.

[7] E. P. Blasch and S. Plano, *Level 5: User Refinement to Aid the Fusion Process*, 5099 SPIE 03, Apr. 2003.

[8] C. A. Shah, P. Watanachaturaporn, M. K. Arora, and P. K. Varshney, *Some recent results on hyperspectral image classification*, IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data, Greenbelt, MD, Oct. 2003.

[9] DataDirect Technologies, *Using Oracle real application clusters (RAC)*, http://www.datadirect.com/techzone/odbc/docs/odbc_oracle_rac.pdf, 2004.

[10] J. C. Lowery, *Scaling-out with Oracle grid computing on Dell hardware*, http://downloadwest.oracle.com/owsf_2003/40379_Lowery.pdf, 2003.

[11] J. A. Gualtieri, S. R. Chettri, R. F. Cromp, and L. F. Johnson, *Support vector machine classifiers as applied to AVIRIS data*, JPL Airborne Geoscience Workshop, Pasadena, CA, Feb. 1999.

[12] S. Tadjudin and D. A. Landgrebe, *Robust parameter estimation for mixture model*, IEEE Trans. Geoscience and Remote Sensing, Vol 38, No. 1, pp. 439-445, 2000.

[13] P. Watanachaturaporn, M. K. Arora, and P. K. Varshney, *Evaluation of factors affecting support vector machines for hyperspectral classification*, American Society for Photogrammetry & Remote Sensing Conf., Denver, CO, May 2004.

[14] L. Biehl and D. Landgrebe, *MultiSpec - A tool for multispectral-hyperspectral image data analysis*, 13th Pecora Symp., Sioux Falls, SD, Aug. 1996.

[15] A. M. Waxman, D. A. Fay, B. J. Rhodes, T. S. McKenna, R. T. Ivey, N. A. Bomberger, and V. K. Bykoski, *Information fusion for image analysis: Geospatial foundations for higher-level fusion*, Intl. Conf. Information Fusion, Annapolis, MD, July 2002, pp. 562-569.

[16] D. A. Fay, R. T. Ivey, N. A. Bomberger, and A. M. Waxman, *Image fusion and mining tools for a COTS environment*, Intl. Conf. Information Fusion, Cairns, Australia, July 2003, pp. 606-613.

[17] S. B. Serpico and L. Bruzzone, *A new search algorithm for feature selection in hyperspectral remote sensing images*, IEEE Trans. Geoscience and Remote Sensing, Vol 39, No. 7, pp. 1360-1367, 2001.

[18] S. Kumar, J. Ghosh, and M. M. Crawford, *Best basis feature extraction algorithms for classification of hyperspectral data*, IEEE Trans. Geoscience and Remote Sensing, Vol 29, No. 7, pp. 1368-1379, 2001.

[19] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, Berkeley Symp. Math. Statist. Prob., 1967, pp. 281-297.

[20] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, *A density-based algorithm for discovering clusters in large spatial database*, Intl. Conf. Knowledge Discovery and Data Mining, Portland, OR, Aug. 1996, pp. 226-231.

[21] A. Hinneburg, C. C. Aggarwal, and D. A. Keim, *What is the nearest neighbor in high dimensional spaces?* Intl. Conf. Very Large Data Bases, Cairo, Egypt, Sept. 2000, pp. 506-515.

[22] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, *Automatic subspace clustering of high dimensional data for data mining applications*, ACM-SIGMOD Intl. Conf. Management of Data, Seattle, WA, June 1998, pp. 94-105.

[23] A. Hinneburg and D. A. Keim, *Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering*, Intl. Conf. Very Large Data Bases, Edinburgh, UK, Sept. 1999, pp. 506-517.

[24] B. L. Milenova and M. M. Campos, *O-Cluster: Scalable clustering of large high-dimensional data sets*, IEEE Intl. Conf. on Data Mining, Maebashi City, Japan, Dec. 2002, pp. 290-297.

[25] B. L. Milenova and M. M. Campos, *Clustering large databases with numeric and nominal values using orthogonal projections*, http://www.oracle.com/technology/products/bi/odm/pdf/ocluster_wnominal_data.pdf, 2003.

[26] G. German, M. Gahegan, and G. West, *Predictive assessment of neural network classifiers for applications in GIS*, Conf. of Geocomputation, Otago, New Zealand, Aug. 1997.