n-Dimensional Visualization

August 12, 2002

Prepared by:

Computer Science Innovations, Inc.
1235 Evans Road
Melbourne, FL 32904-2314

GRAPHICAL ANALYSIS OF COMPLEX DATA

Graphical presentation of data can expedite analysis and interpretation. However, complex problems involve many variables. Standard graphical presentations are constrained by display geometry to at most three variables at a time. This requires the analyst to select – prior to analysis – which variables to ignore in order to reduce complexity of the problem to a displayable level. This "level-set" approach forces the analyst to discard valuable, potentially significant data so that a subset of unknown saliency can be displayed.

CSI has developed a visualization technique based on a unique graphics algorithm that makes possible the display of high dimensional complex datasets. The algorithm, based upon advanced mathematical techniques for the manipulation of affine manifolds in high-dimensional Euclidean spaces, preserves geometric relationships (e.g. connectedness, proximity) and provides user-selected and manipulated perspective views of data occurrence in *n*-dimensional featurespace. This allows analysts for the first time to step into an intuitive depiction of spaces in *n* dimensions (although the algorithm does not inherently limit the number of dimensions that can be modeled, CSI has not implemented displays having more than ten dimensions, due to limitations in display technology and human comprehension). Using Visualizer, the analyst can interact with his data "where it lives". Dimensional depiction is fully spatial: no glyphs, multiple panes, or artificial pseudo-color. Handling data "as it is, where it lives" for the first time often provides fresh insight, true intuition, and accurate understanding.

Insights about data in a particular problem domain, and relationships therein, which may be theoretically predicted, can readily be observed using this technique. The algorithm relies upon the use of carefully selected unitary transformations obtained by the Gram-Schmidt Orthonormalization process. It is highly parallel, computationally efficient, and can be applied to problems in Euclidean spaces of arbitrary dimension.

CSI's *n*-Dimensional Interactive Graphics Visualizer enables the human analyst to select non-intuitive yet pertinent features within a set of complex multi-dimensional data and then observe the resulting data, thus providing greater understanding and allowing the analyst to focus on the objective problem rather than wading through endless, seemingly unrelated data. CSI has applied Visualizer to assist analysts in processing and understanding large amounts of complex or "wide" data, for two primary purposes:

- For the data analyst to construct descriptive models of the data, understand complex relationships among the data, select salient (predictive) features, identify outliers, perform auto-clustering, etc. – as an end goal or as prerequisite to building predictive models.
- For the application or domain analyst – after a predictive model has been built for a specific problem – to quickly spot events-of-interest detected in the data; spot significant patterns of events; assess and prioritize relative importance, inter-dependencies, and significance of the events and conditions; identify newly occurring variant events and relationships; isolate and extract significant clusters of events for detailed analysis.

This paper describes the design and technology of CSI's Visualizer, and notes what further development is required to fully deploy its capabilities beyond its current prototype stage. This description is presented here within an example of its application to the problems of (1) network intrusion detection, and (2) detecting real vs. faux diabetes.

VISUALIZATION BACKGROUND

Over of the past 10+ years, CSI has applied visualization techniques to the analysis of many kinds of data from disparate domains, including radio signal, image, acoustic, optical, financial, and others. Visualization has become a fundamental component of our approach to pattern analysis problems of all types.

The most effective pattern recognition equipment known to man is "grayware": the stuff between your ears. Manual knowledge discovery is facilitated when domain data are presented in a form that allows the human mind to interact with them at an intuitive level. Visual data presentation provides a high-bandwidth, naturally intuitive gate to the human mind.

Data visualization is more than depiction of data in interesting plots. The goal of visualization is to help the analyst gain intuition about the data being observed. Therefore, visualization applications frequently assist the analyst in selecting display formats, viewer perspectives, and data representation schemas that foster deep intuitive understanding.

Several conventional techniques exist for the visualization of 2 and 3-dimensional data: scatter-plots of features by pairs, histograms, relationship trees, bar charts, pie charts, tables, etc. Modern visualization tools often include other capabilities such as drill-down, hyper-links, roam/pan/zoom, animation, morphing, glyphs, etc.

The standard (i.e., level-set) approach to the graphical data fusion problem requires the generation and analysis of multiple displays because it cannot use more than three components at a time. This leads to two serious problems:

1. The number of displays required to present N components by pairs is: $N!/((2!)\ (N-2)!)$ (e.g., if N=6, 15 displays must be examined [and remembered!] to draw inferences using the level-set approach).
2. The level-set approach can mislead users attempting to find correlations in high-dimensional data. Pairwise correlation of the components of a random variable does not insure that the components are jointly correlated.

When data have more than three dimensions, they become more difficult to conceptualize. Graphical techniques for representing high-dimensional data are challenging to implement and use. Visualization is a valuable feature analysis method. Visual analysis of high-dimensional data can be done in several ways:

- Using "glyphs": indicating two features as spatial x-y coordinates, and other features using aliases such as color, shape, size, etc. This simple type of display is often supported by conventional OLAP tools.
- Performing orthographic projection: using spatial dimensions to plot two or three features, while ignoring the rest.
- Performing "modified" stereographic projection: the CSI Visualizer Workstation creates spatial plots in up to 10-dimensions on a single display (without using glyphs, pseudocolor, or non-euclidean coordinate systems).
- Using tree plots, parallel coordinates, dendograms, etc.

If the data are color coded before plotting, the relationships between different categories (e.g., intrusive vs. non-intrusive) can be seen visually.

The Two Dimensionality Models

Visualization is just one of several techniques that may be applied to the analyst efficiency problem. The application examples in this paper are specific to IDS (network Intrusion Detection Systems) and disease prediction; however, the same techniques can also be applied to other problems.

In real-world applications, transaction data (e.g., TCP dump and BSM log files; or patient records) exist as records typically having many fields. This "high-dimensionality" introduces some interesting challenges into the problem of collecting, storing, accessing, visualizing, and analyzing this data. Success in any of these areas requires the formulation of a model to handle this high-dimensionality. There are two such dimensionality models in general use: the data warehouse hypercube model, and the spatial model. More familiar to mathematicians and statisticians than the standard data warehouse hypercube model is the spatial model. In this model, rows in the data set are regarded as vectors in an *n*-dimensional space. The data are organized in this space according to numeric coding of values in the columns. The "meanings" of these columns do not drive the analysis. Rather, it is the spatial relationships of rows in space that determine their significance and utility.

A Visualizer plot showing an abstract data set in three dimensions is shown in Figure 1. The three dimensions available do not provide sufficient information to elucidate the structure of the population, or distinguish between the two classes of data ("red" and "green").

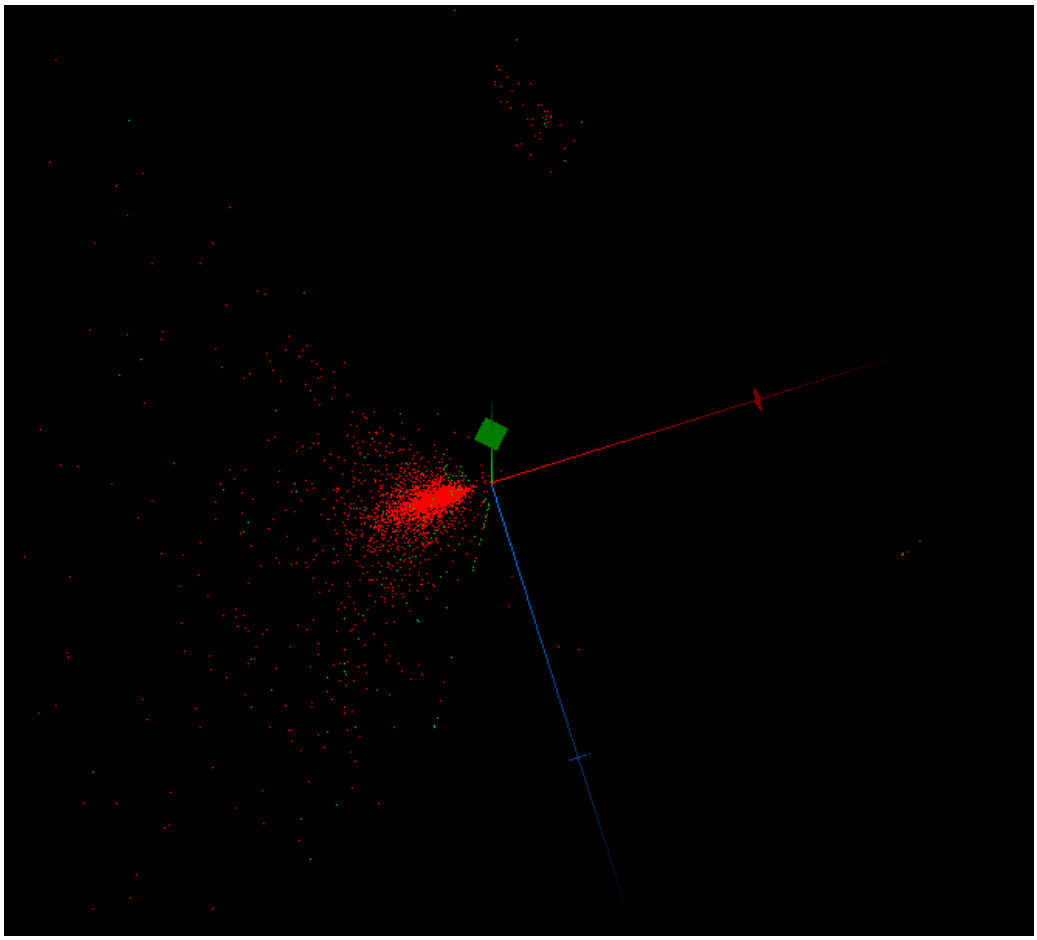

Figure 1. An abstract data set in three dimensions.

Next, by incorporating five additional dimensions of data, and using the Visualizer to plot the data in 8-dimensional space, much more structure and discrimination between the two classes of interest are seen in Figure 2 below.

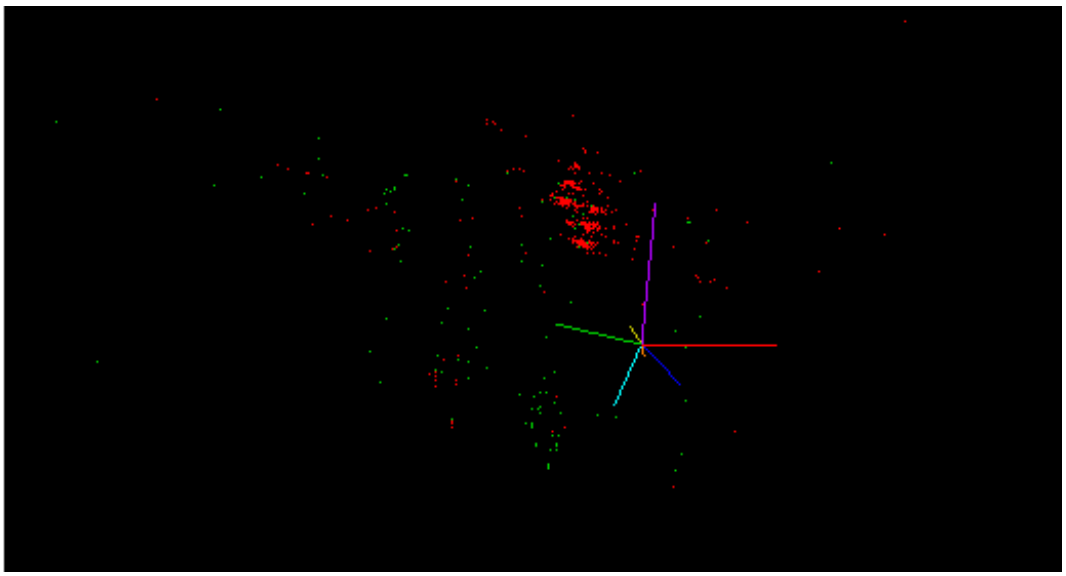

Figure 2. An abstract data set in 8-dimensional space.

APPLICATION OF VISUALIZATION TO IDS APPLICATIONS

Intrusion detection systems monitor network traffic to detect patterns consistent with intrusion behaviors (an attack as well as precursors). These patterns are complex asynchronous time-series of many nominal and numeric factors. The patterns are embedded in a large volume of irrelevant (non-intrusive) data. The volume and variability of these data make the identification, characterization, and exploitation of latent patterns very difficult. Further, because the data are naturally high dimensional, it is difficult for analysts to develop an intuitive understanding of latent patterns, complicating analysis.

There are some well-known techniques for visualizing complex high-dimensional data (for example, Inselberg plots); many of these do not depict the data in a way that makes its natural spatial characteristics (e.g., shape, structure) intuitive.

CSI's visualization paradigm supports the display of more than three dimensions simultaneously in such a way that important intuitive aspects of spatial characteristics are preserved. Such visualization methods, used in combination with other high-end data mining tools, have the potential to assist in the analysis of network intrusion detection by making the complex patterns associated with intrusion behaviors more accessible to analysts.

Network Metrics

Much of IDS analysis and related operational work requires the collection and analysis of network operational statistics, called network metrics. Network metrics are measurements that allow administrators to monitor and optimize network performance. These measurements are performed, aggregated, and analyzed by metric applications. Metric applications can be distributed or centralized.

A critical aspect of IDS analysis is the development of support tools and automation that leverage the overloaded human analyst. Visualizer organizes large amounts of complex data in an intuitive visual form, in which one display can represent over 25,000 pages of TCPdump. This aids the analyst in efficiently recognizing significant patterns that might otherwise remain undetected, or require months of effort to discover and characterize. Visualizer can also serve the operational user by presenting network metric data in an easily understood visual format. In this way, complex multi-faceted data can be understood quickly and intuitively.

Network metrics are of two general types**:** Historic and Prognostic.

Historic

Historic metrics index and count network events, such as number of messages/packets sent, type and number of requests serviced by node, traffic loading as a function of time, successful/unsuccessful service requests, etc. These indexed measurements are generally made available to the intrusion analyst or other network professional through OLAP-like (online analytical processing) tools such as Excel, Seagate Analysis, etc. These tools allow organization, tabulation, and visualization of these data, giving insight into current network operation. Visualization is a critical component of this analysis.

Prognostic

Prognostic metrics use applications with embedded intelligence to interpret network measurements. Interpretation includes recognition of significant patterns in network traffic, intelligent trend analysis, prediction of single and multiple points of failure, and execution of "what-if" scenarios. These analyses are made available to the network manager by automatically generated reports, which can be greatly enhanced by the inclusion of appropriate visualizations of complex network behaviors. Prognostic metrics using sophisticated visualization support reliable identification of "out of nominal" conditions, and sometimes give deep insight into network evolutionary behaviors.

Prognostic metrics supported by sophisticated visualization provide actionable information for:

- detection and monitoring of network compromise (e.g., anomalous activity, intrusion detection)
- network planning and growth modeling
- prediction of failure points and loading bottlenecks

Prognostic metrics with visualization allow the intrusion analyst to generate insightful views of network operation, fostering deep intuition regarding patterns in network behaviors.

IDS Analyst Visualization Examples

A visualization of TCP Dump data using the CSI Visualizer appears in Figure 3 below. This display shows an intrusion event during which an attacker using a stolen password TELNETS into a network and exhibits behavior characteristic of a hacker who has gained root shell access through the firewall. Normal network behaviors are the green squares seen near the origin; suspicious behaviors are in yellow, and compromising behaviors are in red. There is clear visual separation between the non-intrusive (green) and suspicious/intrusive behaviors (yellow/red).

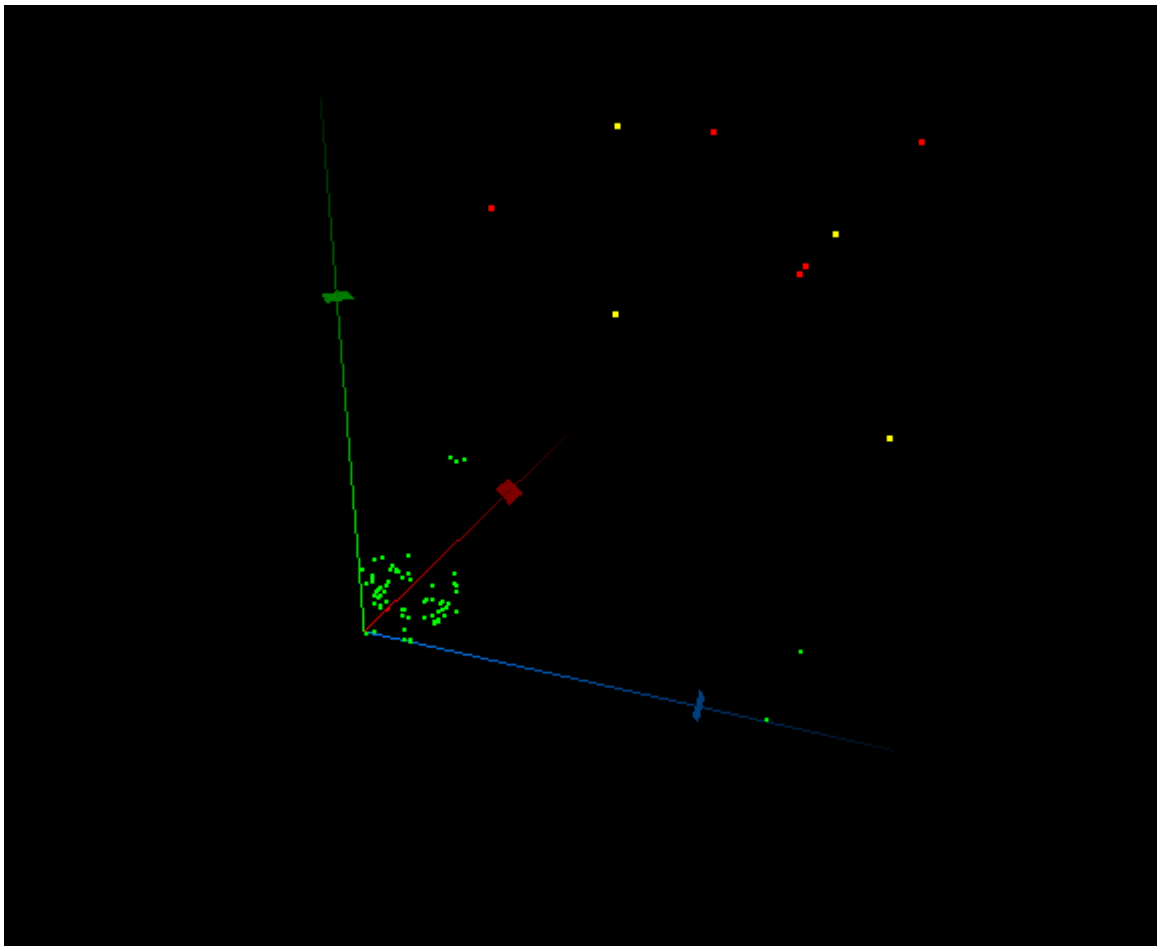

Figure 3. Visualization of the TCP Dump data using the CSI Visualizer

Through experience in processing vast amounts of network data, CSI has discovered that different types of network activity will present uniquely-appearing shapes or patterns in the display, which are not always visible until the user copters the axes around to the plane in which the pattern aligns. CSI has also learned that standard visualization displays restricted to fewer dimensions typically obscure or obviate the pattern shapes which reveal significant insights in Visualizer. Such unique shapes help the analyst readily identify the *type* of attack underway, and thereby prioritize his response relative to degree of threat or potential damage therefrom.

Another visualization technique mentioned earlier is the Inselberg plot which is a "parallel coordinate" plot. Rather than plotting one dimension on a single vertical axis as is usually done, parallel coordinate plots use a separate vertical axis for each dimension (hence the name "parallel coordinates"). A point in n-dimensional space becomes a "trace" having n nodes.

Certain types of intrusive behavior have a different visual appearance from non-intrusive behaviors as can be seen in the following Inselberg Plot, Figure 4 (the same data plotted above). Non-intrusive behavior is the upper plot, and intrusive behavior is the lower plot. In the plot you will notice that there are more non-intrusive behaviors than intrusive ones, and that while the two classes generally look different, there are certain similarities. This plot therefore tells us that the features we have selected for analyzing this behavior discriminates between the two classes.

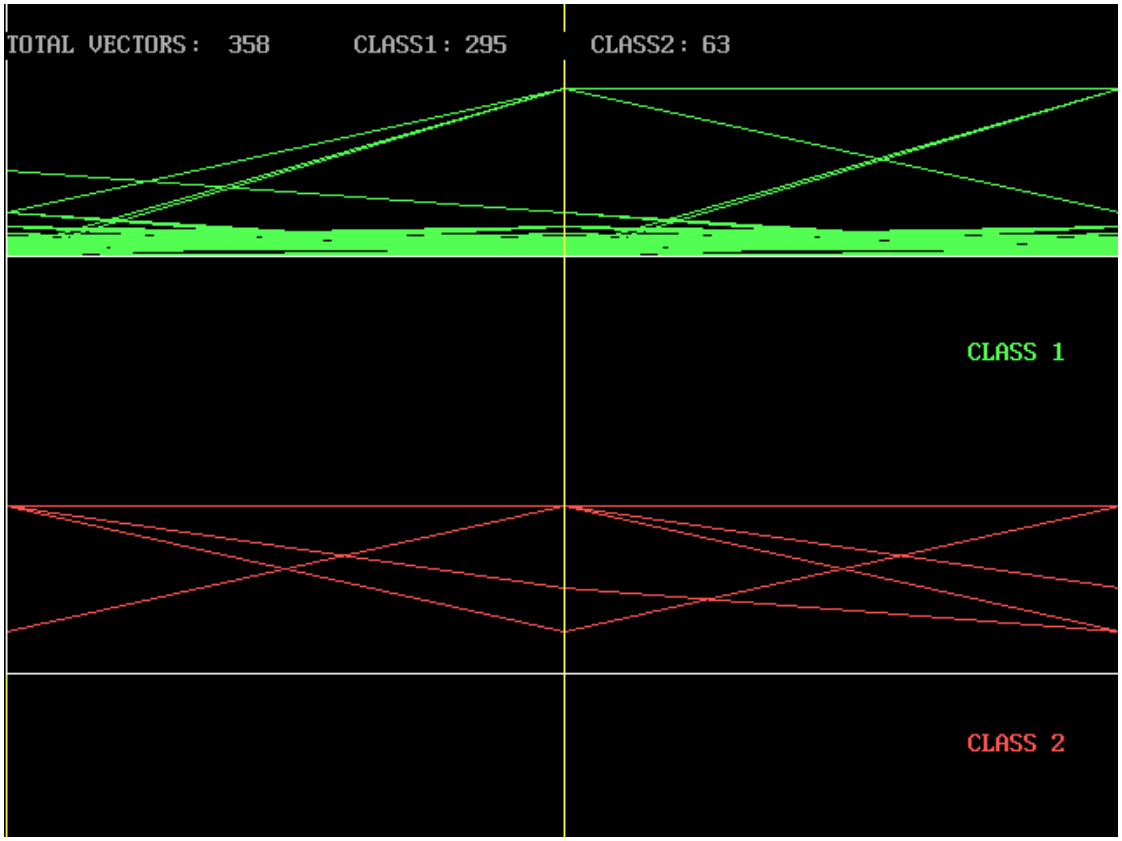

Figure 4. Inselberg Plot of the same TCP Dump data

Benefits of *n*-Dimensional Visualization for IDS Analysis

1.) Allows the IDS analyst to develop "intuition" about network traffic and behavior. When an analyst is able to see data, he develops an intuitive understanding of it that can be obtained in no other way.

2.) Can lead to the direct discovery of network behavior "rules."

3.) An effective way to scan large amounts of data for "significant differences" between classes. If the icons representing suspicious network threads are well separated from trusted threads along some axis, this will be visually apparent. The feature that axis represents, then, accounts for some of the difference between normal and intrusive behaviors.

4.) Detects poorly conditioned data. When data are plotted, errors in data preparation often show up as unexpected irregularities in the visual pattern of the data.

5.) Helpful in selecting a data model. For example, data that is normally distributed will have a characteristic shape. Visual analysis lets the analyst "see" which of several models might be most reasonable for a population being studied.

6.) Helps spot outliers and missing data. When data is visualized, outliers (data not conforming to the general pattern of the population) are usually easy to pick out. Missing records may show up as "holes" in a pattern, and missing fields within a record often break patterns in easily detectable ways.

VISUALIZATION: REAL VS. FAUX DIABETES

The first part of the problem of detecting real vs. faux diabetes requires feature analysis. The purpose of the analysis is to take a "quick look" for clustering in a high-dimensional feature space to help validate the diabetes episode. Do the defining and excluding codes in the episode correlate with other claim factors?

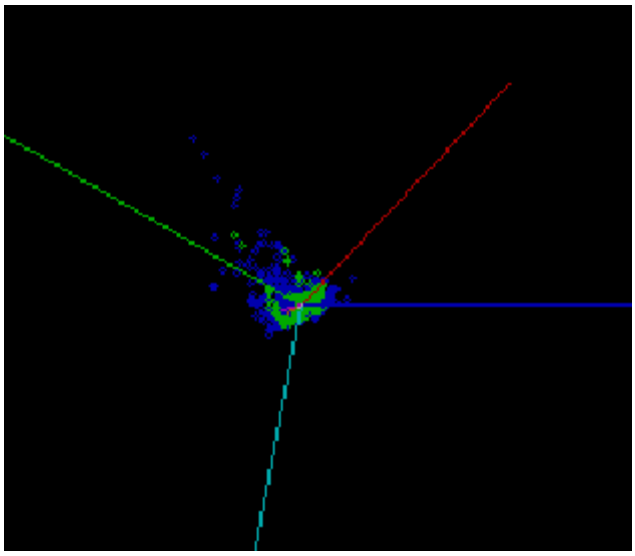

FIGURE A

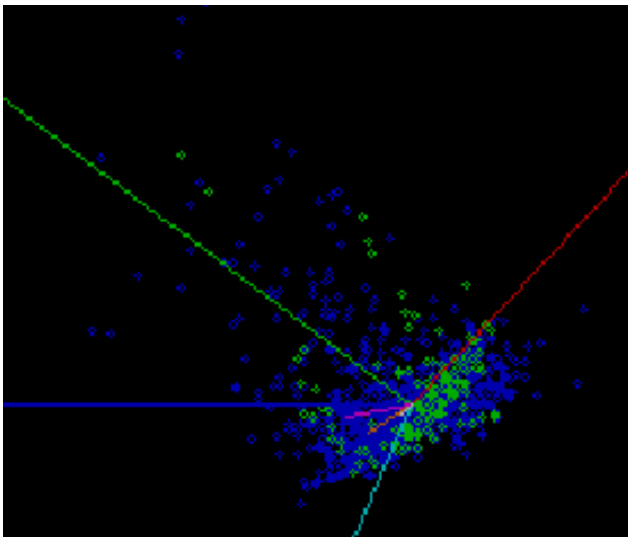

FIGURE B

The Real diabetes claims are depicted in blue, the faux claims in green.

As we approach the population, some structure begins to become apparent. Figure A shows some "tailing" of the population to the upper left, and Figure B shows a mid-field concentration of faux claims. Some peeling off of a blue aggregation is barely visible from this perspective.

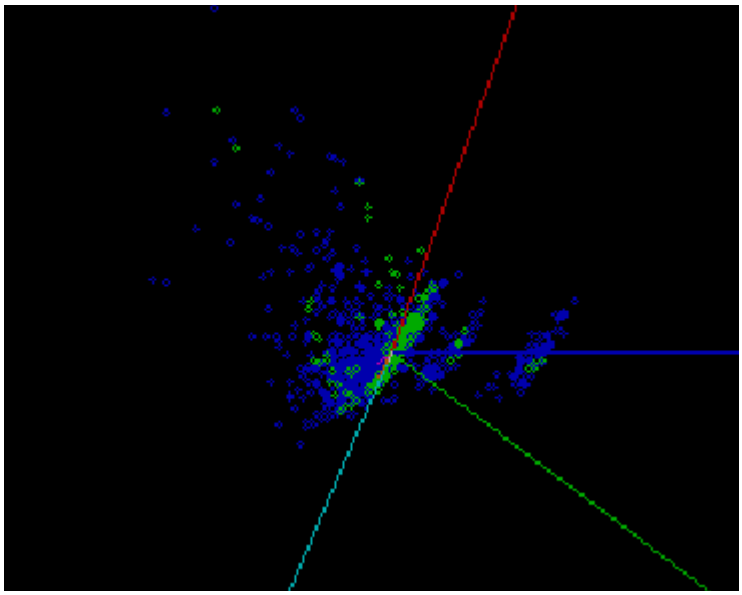

FIGURE C

Piloting to a different perspective (Figure C), two class-homogeneous aggregations are seen to the right of the main body. These are claims associated with surgical procedures; one represents male patients and the other represents female patients.

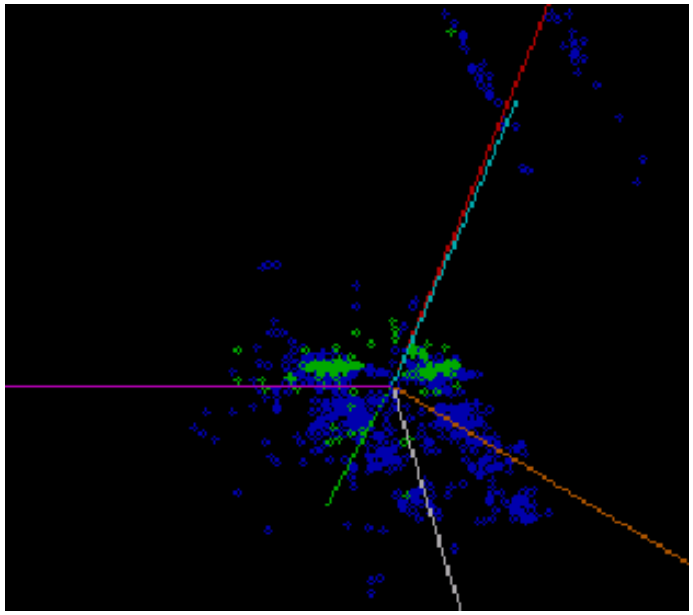

FIGURE D

Swinging around to a new perspective (Figure D), it is seen that the faux claims consist of two distinct linear aggregations, while the real claims are well separated. This indicates that this feature set does a reasonable first-cut discrimination between the two classes.

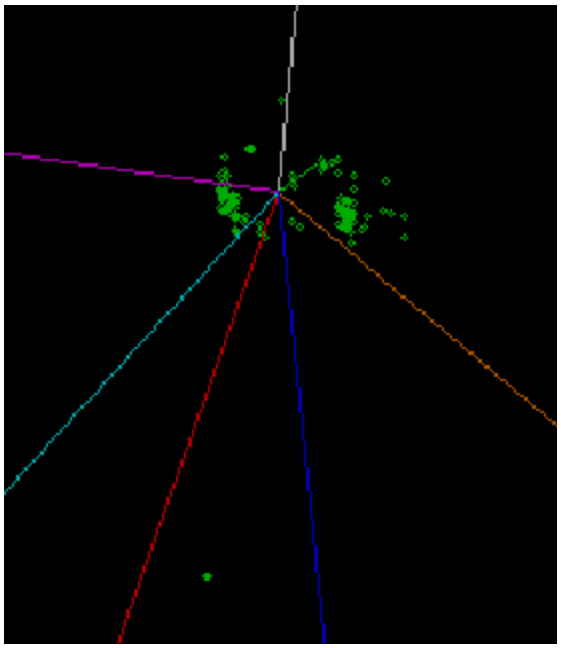

FIGURE E

In Figure E, the real claims have been suppressed, showing clearly the bipartite structure of the faux claims in this representation.

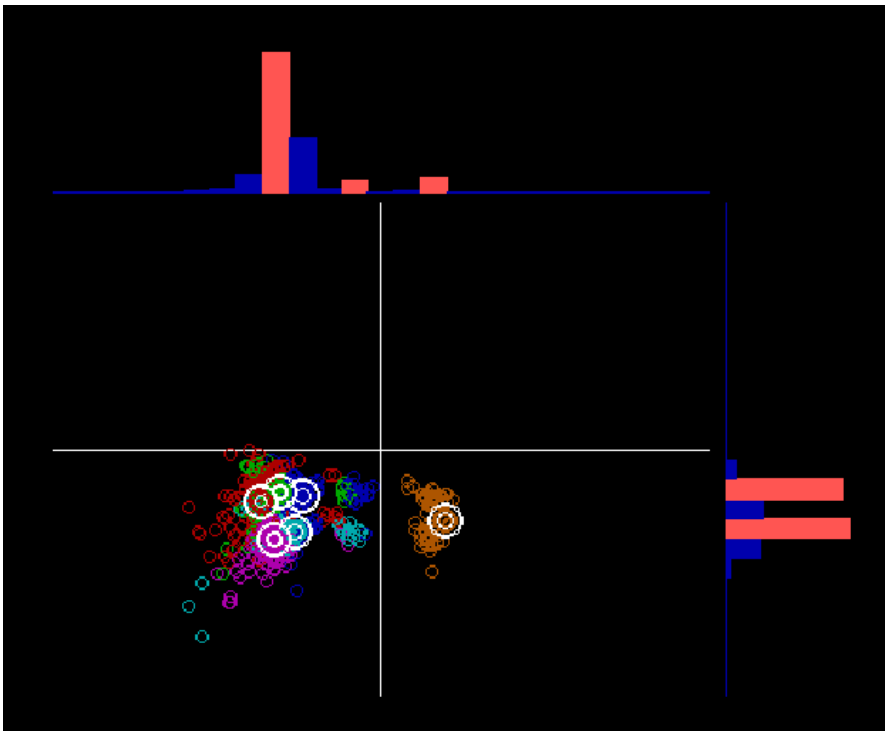

FIGURE F

To determine the significance of the two outlying aggregations, CSI technology is applied to perform unsupervised clustering. The outliers In Figure F are seen to consist of pairs of components. The outlying "surgical" clusters are written to a file for individual analysis. They are found to have the same distribution of real and faux records as the population, indicating that they carry little clinically significant discriminating information. The presence of visual segregation between real and faux diabetes in this feature set suggests that the defining and excluding codes in the diabetes episode do correlate with other claim factors. It might be possible to use these factors to build a population model for discrimination.

Visualizer and Other Methods

CSI's Visualizer is currently in a prototype stage. A description of its current capabilities, as well as a listing of design-intentioned capabilities, is provided later in this paper.

In addition to the Visualizer, CSI has a number of other visualization methods (e.g., orthographic projection, Inselberg plots). During a typical visualization task, several such applications can be applied to the data in question. The method giving the most effective representation for the problem at hand is used during subsequent analysis.

Whether visualization techniques are effective depends upon the problem at hand and the data representing it. It is difficult to predict how effective sophisticated visualization will be in a particular problem domain without actually applying it to some sample data. Furthermore, the relative value of visualization depends on the strength of underlying algorithms for data preparation, feature selection and enhancement, and pattern recognition – this is where CSI's next-generation technologies excel.

Geometry preserving visualization methods (such as those used by the CSI Visualizer) work best when the data are entirely numeric (no discrete features). They are less effective when discrete features are present. CSI has data transformation processes to convert other types of data to enable effective visualization.

Sophisticated visualization techniques work best when used in conjunction with other advanced methods, such as adaptive clustering. The effective application of these methods usually requires a high level of expertise and specialized analytic skills on the part of the developer. However, Visualizer users typically become adept at using it rather quickly. No special mathematical skills are required to use it, although some of the functions (e.g., chi-square) require special knowledge to interpret.

CSI VISUALIZER BENEFITS

- All data are exploited in a single display.
- False indications of correlation are mitigated.
- Low-frequency data occurrence – often the most critical – can easily be identified and correlated.
- Because all parameters are used, even data that overlap in several parameters can be discriminated.

CSI VISUALIZER FEATURES

Viewing Features

The Visualizer's proprietary *n*-dimensional graphics algorithm creates geometrically faithful views of data points as they actually reside in their high-dimensional space. Nearness, occlusion, and perspective are preserved.

The Copter feature allows the user to "fly around," even through, the data, viewing it from different locations in n-dimensional space. Auto-centering keeps the data centered in the field of view.

Data can be pseudo-colored by subset membership or a user supplied attribute ("ground truth").

Up to ten named, color-coded axes indicate the spatial dimensions being displayed.

The mouse can be used to select subsets of data right on the screen by enclosing them in a rectangle (Lasso function).

Selected subsets are named, and highlighted in color. Subset membership is a persistent attribute (e.g., preserved under "coptering").

Analysis Features

Statistics

The Visualizer gives the minimum, maximum and average values assumed by the components of the points for each defined subset.

Autoclustering (Autocluster function) may be selected to let the Visualizer automatically group data into a user-determined number of clusters based upon the relative distribution of data. This clustering is performed in up to ten dimensions simultaneously.

Histogramming (Stats function) gives the relative frequency of data values for each defined subset.

Aggregate Description (KBT function) uses conventional measurements and the chi-square statistic to descriptively compare any ground-truth class with the entire population.

Visualizer is highly scalable, has limited computational requirements, and does not need a "supercomputing environment" in order to function effectively.

Automatic Perspective Finder

An adaptive routine has been incorporated into the prototype version of Visualizer that allows the analyst to specify a pair of classes (or "all"). The routine then moves the viewer's perspective to a location that optimizes visual separation between the specified classes, while the viewer watches. This is very helpful for the analysis of complex situations, where the selection of a good viewing perspective in high-dimensional space is difficult. For example, finding a good discriminating perspective for a sample containing both intrusive and non-intrusive behaviors could provide important insight into certain types of intrusive behaviors.

Toolkit Implementation

Further development and extension of Visualizer into a re-usable toolkit can easily be achieved utilizing existing CSI routines (from our Advisor Toolkit) that support high-end analysis functions.

- Regression Tools
  - Feature Analyzer (bayesian)
  - PCA (Karhunen-Loeve)
  - Neural Networks (Radial Basis Function, Multi-Layer Perceptron)
  - Hybrid Classifiers
  - Performance Boosters
- RML Tools
  - Automatic Rule Discovery
  - Super Lexical Processing
    - Multi-factor sorting and searching
    - Parsing/pattern matching
    - Report Generation

Specific Design-Intentioned Development Objectives

Usability Enhancements

Enhancements to make Visualizer more effective and user-friendly:

1) Filter and/or Display unknown and confused vectors in separate colors.
2) Color code the axis control buttons.
3) Display, capture, and enter the coordinates for each axis that defines the perspective.
4) Set perspective waypoints so the user can get back to a known perspective.
5) Limited "Grab and View"
6) Limited "Grab and Tag"

Display Techniques

Identify improved Visualization techniques to make the display more useful to the analyst. For example, currently, only ten dimensions are displayed because Principal Component Analysis (PCA) is performed on the 43 features provided by the Cognitive Intrusion Detection System. Instead of performing PCA, giving an independent axis to each feature may be much more useful to the operator. This phase of development will also explore methods for displaying data in real time. For example, this may involve showing newer sessions with greater brightness, fading over time. These enhancements will make the Visualizer more effective and also give it a real-time capability.

Development Goals for use with IDS:

Filter/Display Associated Traffic

Development in this phase would identify the sessions associated with each other (e.g. for IDS, by protocol, by IP address, etc.), allow the filtering of these related sessions, and display them in a unique manner (color coding, "connect-the-dots", etc.). These enhancements will give the operator/analyst a greater understanding of the macro-behaviors in the data universe (e.g. for IDS, the defended network).

Integrate Visualizer with the Cognitive Engine GUI

The design intention is to integrate the Visualizer engine with the problem-domain-specific Cognitive Engine (e.g. Cognitive Intrusion Detection System). The enhancements include:

1) Full implementation of "Grab and View" so that operator/analyst may move seamlessly among the multiple data representations (for IDS: Packets, Alarms, Visualized sessions)
2) Full implementation of "Grab and Tag" so that operator/analyst may move seamlessly among the multiple data representations
3) Implement a single integrated GUI.