OPEN DATA SCIENCE CONFERENCE

San Francisco | Oct. 28 - Nov. 1, 2019
TUTORIAL

INTERPRETABLE KNOWLEDGE DISCOVERY REINFORCED BY VISUAL METHODS
TUTORIAL:
INTERPRETABLE KNOWLEDGE DISCOVERY
REINFORCED BY VISUAL METHODS

Prof. Boris Kovalerchuk
Dept. of Computer Science,
Central Washington University, USA
http://www.cwu.edu/~borisk
Overview

- This tutorial covers the state-of-the-art research, development, and applications in the area of interpretable knowledge discovery reinforced by visual methods.

- The topic is interdisciplinary bridging of scientific research and applied communities in Machine learning, Data Mining, Visual Analytics, Information Visualization, and HCI.

- This is a novel and fast growing area with significant applications, and potential.

- These studies have grown under the name of visual data mining.

- The recent growth under the names of deep visualization, and visual knowledge discovery, is motivated considerably by:
  - deep learning success in accuracy of prediction and
  - its failure in providing explanation/understanding of the produced models without special interpretation efforts.

- In the areas of Visual Analytics, Information Visualization, and HCI, the increasing trend toward machine learning tasks, including deep learning, is also apparent.

- This tutorial reviews progress in these areas with analysis of what each area brings to the joint table.
Human Visual and Verbal generalization vs. Machine Learning generalization for Iris data

Examples of Setosa, Versicolor and Virginica Iris petals.

Setosa – small length and small width of petal
Versicolor – medium length and medium width of petal
Virginica - large length and medium to large width of petal

Example of logistic regression classification of these classes of Iris [Big data, 2018].


GLC-L
Classes 2 and 3 Errors 3
Wrong ML prediction (overgeneralization) in this example is not a result of insufficient training data.

It is a result of the task formulation – search for the linear or non-linear discrimination lines in LDA, SVM and DT that classify every point in the space to one of these three classes.

Instead we can use envelopes around training data of each class – small length and width of petal for setosa. Points outside of the envelopes are not recognized at all. The algorithm refuses to classify them.

How can we know that we need to use an envelope not such functions for n-D data?

We need visualization tools to visualize n-D data in 2-D without loss of n-D information (lossless visualization) allowing to see n-D data as we see 2-D data.

Visualization will allow to see an n-D granule that corresponds to small petal in linguistic terms – fuzzy sets or subjective probabilities to formalize these linguistic terms.

Synergy of computing with images [Kovalrechuk, 2013].

Kovalrechuk, B., Quest for rigorous intelligent tutoring systems under uncertainty: Computing with Words and Images, In: Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013. pp. 685 - 690, DOI: 10.1109/IFSA-NAFIPS.2013.6608483
Overview

The tutorial covers five complementary approaches:

1. to visualize ML models produced by analytical ML methods,
2. to discover ML models by visual means,
3. to explain deep and other ML models by visual means,
4. to discover visual ML models assisted by analytical ML algorithms,
5. to discover analytical ML models assisted by visual means.

What is not covered

- SAS visual data mining and machine learning that is a visual interface for all analytical ML steps that integrates analytical processes.

The target audience

- Data science researchers, graduate students, and practitioners with the basic knowledge of machine learning.

Sources: Multiple relevant publications including presenter’s books:

- “Visual and Spatial Analysis: Advances in Visual Data Mining, Reasoning, and Problem Solving” (Springer, 2005), and
Dr. Boris Kovalerchuk

- Dr. Boris Kovalerchuk is a professor of Computer Science at Central Washington University, USA. His publications include three books "Data Mining in Finance" (Springer, 2000), "Visual and Spatial Analysis" (Springer, 2005), and "Visual Knowledge Discovery and Machine Learning" (Springer, 2018), a chapter in the Data Mining Handbook and over 170 other publications.

- His research interests are in data mining, machine learning, visual analytics, uncertainty modeling, data fusion, relationships between probability theory and fuzzy logic, image and signal processing.

- Dr. Kovalerchuk has been a principal investigator of research projects in these areas supported by the US Government agencies. He served as a senior visiting scientist at the US Air Force Research Laboratory and as a member of expert panels at the international conferences and panels organized by the US Government bodies.

- http://www.cwu.edu/~borisk
Highlights

• Enhancing visualization and machine learning for discovering hidden understandable patterns in multidimensional data (n-D data).
  – The fundamental challenge -- we cannot see multidimensional data with a naked eye and need visual analytics tools (“n-D glasses”). It starts at 4-D.
• Often we use non-reversible, lossy dimension reduction methods such as Principal Component Analysis that convert, say, every 10-D point (10 numbers) to 2-D point (two numbers) in visualization.
  – They are very useful, but they can remove important multidimensional information before starting discovering complex n-D patterns.
• The hybrid methods combine advantages of reversible and non-reversible methods for knowledge discovery in n-D data.
• We will review reversible and non-reversible visualization methods and combine them with analytical ML/DM methods for knowledge discovery.
**Black box vs. White box**

**Black Box Models**
- Deep learning Neural networks
- Boosted trees
- Random forests
- ...

Often less interpretable but more accurate.

**Glass Box Models**
- Single Decision Trees
- Naive-Bayes
- Bayesian Networks
- Propositional and First Order Logic rules

Often less accurate but more interpretable and human understandable.

**Often we are forced to choose between accuracy and interpretability. It is a major obstacle to the wider adoption of Machine Learning in areas with high cost of error**, such as in cancer diagnosis, and many other domains where it is necessary to **understand, validate, and trust decisions**. Visual Knowledge discovery helps to get both model **accuracy and its explanation**.

APPROACHES
Approaches to visualize Machine Learning (ML) models produced by the analytical ML methods

- These visualization of ML models are used
  - to demonstrate and understand ML models such as CNN, Association Rules and others,
  - to show the superposition of input data/images with heat maps of model layers,
  - dataflow graph in Deep Learning models,
  - differences in the internal representations of objects, adversarial cases and others using t-SNE and other visualization methods.
Demonstration of DNN models to understand them via heat maps visualizing discovered features

ConvNets: Multiple Trainable Layers, Hierarchical Representations

Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor

Deep Learning: Representations are hierarchical and trained

Video

https://www.youtube.com/watch?v=VsnQf7exv5I
Visualization of Association Rules

- Example: rule {onions, potatoes} ⇒ {burger}
  - If customers buy both onions and potatoes, they are more likely to buy a burger.
- Association rules – interpretable
- Challenges –
  - limited specifics (e.g., why rule confidence is less than 100%?)
  - incomplete understandings of relation strength
  - How to identify uncertainty of patterns, outliers.
  - Deleting many non-interesting rules.
- Quality Measures:
  - Support -- how frequently an itemset appears in the dataset.
  - Confidence, conf(X ⇒ Y) = supp(X&Y)/supp(X)-- fraction of transactions with X and Y.
  - Lift, lift(X ⇒ Y) = supp(X & Y)/(supp(X) × supp(Y))-- the ratio of the observed support to that expected if X and Y are independent.

Visualization approaches:
- Matrix with rows as Left Hand Side, LHS itemsets of rules and columns as Right Hand side (RHS) itemsets of rules.
- Parallel sets

Challenges:
- Scalability for many LHS and RHS
- readability of small cells having many categories in variables.

Visualization of Association Rules (AR) using Parallel sets for categorical data

- **Approach:**
  - Discovering AR.
  - Deleting dimensions irrelevant to AR.
  - Feeding rules to two **coordinated rule visualizations** (called ARTable and ParSets),

- **User interactions:**
  - Visually explore rules in ARTable
  - Find interesting rules, dimensions, and categories in ARTable
  - Create and optimize the layout of ParSets
  - Validate interesting rules
  - Explore details of rules in ParSets using domain knowledge

**Before**

**After**


ParSets displays dimensions as adjacent **parallel axes** and their values (categories) as **segments** over the axes (points in Parallel Coordinates [Inselberg, 2009]). Connections between categories in the parallel axes form **ribbons** (lines in parallel coordinates). The ribbon crossings lead to visual **clutter**.

Titanic example: Clutter 16.86% – alphabetical ordering of dimensions and categories.

Clutter 11.71% – ordering by the number of grades (categories) in each coordinate.
A: Association Rule Table (ARTable) – the rules with sorting panel. Columns – dimensions extracted from association rule result. Rows – rules with cells indicating the itemsets in the rule. Color intensity expresses % or rule support.

B: ParSets show dimensions in the order displayed in the ARTable. The dimensions in the clicked rule are sorted with the outcome dimension: Edible is on the top. Enabling the filter of “Rule-related category”, Categories absent in ARTable are greyed out.

Green: Edible = Yes, magenta: Edible = No (Poisonous).

Top of the tabular view -- a strong rule between odor and edible mushrooms (odor = none) ⇒ {edible = Yes}).

ParSets places the identified dimension odor right under the top outcome dimension edible

Outlier of the rule is highlighted in the ParSets – a small magenta connection in the odor = none emerges

Measure the level of clutter. Idea – deviation from going straight

https://ars.els-cdn.com/content/image/1-s2.0-S2468502X1930021X-mmc1.mp4
Visualizing dataflow graphs of deep learning models in Tensorflow

- Problem:  
  - Difficulties to optimize Deep Neural Networks (DNN) models due to lack of understanding of how the models work (Black-box problem).

- Approach:  
  - Visualization of NN dataflow

- Goals:  
  - monitor learned parameters and output metrics (scalar values, distribution of tensors for images, audio, etc.)
  - help train and optimize NN models.
  - help understand the structure of dataflow graphs of arbitrary NN at the low-level.

- Tools:  
  - Graph Visualizer,
  - TensorBoard, TensorFlow’s dashboard
  - Olah’s interactive essays
  - ConvNetJS,
  - TensorFlow Playground
  - Keras, MXNet (visualize model structure at the higher-level than TensorFlow)

Approaches to discover ML models by visual means

- Discovering ML models by visual means (lossless/reversible and lossy visual methods for n-D data representation based on Parallel and Radial Coordinates, RadVis, Manifolds, t-SNE General Line Coordinates, Shifted Paired Coordinates, Collocated Paired Coordinates, and others).
We are moving from *visualization of solution to finding solution visually*

- **Why Visual?**
  - To leverage human perceptual capabilities
- **Why interactive?**
  - To leverage human abilities to adjust tasks on the fly
- **Why Machine Learning?**
  - To leverage analytical discovery that are outside of human abilities.
  - We cannot see patterns in multidimensional data by a naked eye.
Visuals for creative thinking

• Scientists such as
  – Bohr, Boltzmann, Einstein, Faraday, Feynman, Heisenberg, Helmholtz, Herschel, Kekule, Maxwell, Poincare, Tesla, Watson, and Watt
• have declared the fundamental role that *images* played in their *most creative thinking*
  Thagard & Cameron, 1997; Hadamard, 1954; Shepard & Cooper, 1982].
• *Albert Einstein: The words or the language, as they are written or spoken, do not seem to play any role in my mechanism of thought.*
Chinese and Indians knew a visual proof of the Pythagorean Theorem in 600 B.C. before it was known to the Greeks [Kulpa, 1994]. Below on the left

\[(a+b)^2 \text{ (area of the large square)} = a^2+b^2+ab+ab=(a+b)^2\]

\[a^2+b^2=(a+b)^2 \text{ (area of the large square)} - 2ab \text{ (4 light green triangles)} = c^2 \text{ (area of inner darker green square)}\]

Thus we follow them-- moving from *visualization of solution* to *finding solution visually with modern data science tools.*
Example of visual knowledge discovery in 2-D

The common guess without visualizing data is to try a simplest linear discrimination function (black line) to separate the blue and red points. It will obviously fail.

In contrast a quick look at these data, immediately gives a visual insight of a correct model class of “crossing” two linear functions, with one line going over blue points, and another one going over the red points.

These “crossing” classes cannot be discriminated by a single straight line. Each single projection does not discriminates them. Projections overlap.
How to reproduce the success in 2-D for n-D data?

In high-dimensions we cannot see data with a naked eye. We need methods for lossless and interpretable visualization of n-D data in 2-D.
The goal of multivariate, multidimensional visualization is representing \textit{n-tuples} (\textit{n-D points}) in 2-D or 3-D, e.g., (3,1,0.5, 7, 1.19, 0.13, 8.1)

Often multidimensional data are visualized by \textit{lossy} dimension reduction (e.g., PCA) or by \textit{splitting} \textit{n-D data} to a set of low dimensional data (pairwise correlation plots).
  
  – While splitting is useful it destroys integrity of \textit{n-D data}, and leads to a \textit{shallow understanding} complex \textit{n-D data}.

To mitigate splitting difficulty
  – an additional and difficult perceptual task of \textit{assembling} 2-dimensional visualized pieces to the whole \textit{n-D record} must be solved.

An alternative way for deeper understanding of \textit{n-D data} is
  – developing visual representations of \textit{n-D data} in low dimensions \textit{without such data splitting and loss of information} as \textit{graphs not 2-D points}.
  – E.g., Parallel and Radial coordinates.
  – Challenge -- occlusion
WBC data

Benign and malignant cancer cases overlap. Interpretation of dimensions is difficult. Non-reversible lossy methods: 9-D to 2-D.

Theoretical limits to preserve n-D distances in 2-D: Johnson-Lindenstrauss Lemma

- This lemma implies that only a small number of arbitrary n-D points can be mapped to k-D points of a smaller dimension $k$ that preserve n-D distances with relatively small deviations.
- Reason: the 2-D visualization space does not have enough neighbors with equal distances to represent the same n-D distances in 2-D.
- Result: the significant corruption of n-D distances in 2-D visualization.
Lemma [Johnson, Lindenstrauss, 1984].

Given $0 < \varepsilon < 1$, a set $X$ of $m$ points in $\mathbb{R}^n$, and a number $k > 8\ln(m)/\varepsilon^2$, there is a linear map $f : \mathbb{R}^n \to \mathbb{R}^k$ such that

\[(1 - \varepsilon) \|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \varepsilon) \|u - v\|^2\]

for all $u, v \in X$.

In other words, this lemma sets up a relation between dimension $n$ and a lower dimension $k$ and the number of points $m$ of these dimensions when the $n$-D distance can be preserved in $k$-D with some allowable error $\varepsilon$.

How often have you seen this lemma that exists over 30 years in the ML textbooks?
Versions of Johnson-Lindenstrauss lemma

- Defines the possible dimensions $k < n$ such that for any set of $m$ points in $\mathbb{R}^n$ there is a mapping $f: \mathbb{R}^n \to \mathbb{R}^k$ with “similar” distances in $\mathbb{R}^n$ and $\mathbb{R}^k$ between mapped points. This similarity is expressed in terms of error $0 < \varepsilon < 1$.

- For $\varepsilon = 0$ these distances are equal. For $\varepsilon = 1$ the distances in $\mathbb{R}^k$ are less or equal to $\sqrt{2} S$, where $S$ is the distance in $\mathbb{R}^n$. This means that distance $s$ in $\mathbb{R}^k$ will be in the interval $[0, 1.42S]$.

- In other words, the distances will not be more than 142% of the original distance, i.e., it will not be much exaggerated. However, it can dramatically diminish to 0. The exact formulation of this version of the Johnson-Lindenstrauss lemma is as follows.

- **Theorem 1** [Dasgupta, Gupta, 2003, theorem 2.1]. For any $0 < \varepsilon < 1$ and any integer $n$, let $k$ be a positive integer such that

$$k \geq 4(\varepsilon^2/2 - \varepsilon^3/3)^{-1}\ln n$$

then for any set $V$ of $m$ points in $\mathbb{R}^k$ there is a mapping $f: \mathbb{R}^n \to \mathbb{R}^k$ such that for all $u, v \in V$

$$(1 - \varepsilon)||u-v||^2 \leq ||f(u)-f(v)||^2 \leq (1 + \varepsilon)||u-v||^2$$

- A formula (3) from [Frankl, Maehara, 1988] states that $k$ dimensions are sufficient, where

$$k = \left\lceil 9(\varepsilon^2 - 2\varepsilon^3/3)^{-1}\ln n \right\rceil + 1$$
Theoretical limits to preserve n-D distances in 2-D: Johnson-Lindenstrauss Lemma

Johnson-Lindenstrauss Lemma shows that to keep distance errors within about 30% for just 10 arbitrary high-dimensional points, we need over 1900 dimensions, and over 4500 dimensions for 300 arbitrary points. Visualization methods do not meeting these requirements.
TensorFlow Embedding projector and warnings

TensorFlow equipped with a visualization module -- Embedding Projector with PCA and t-SNE (t-distributed stochastic neighbor embedding)

**Warnings** [van der Maaten, 2018; Embeddings, 2019].
- PCA can **distort local neighborhoods**.
- t-SNE can **distort global structure** trying to preserve local data neighborhoods
- t-SNE may not help to find **outliers** or assign meaning to **densities** in clusters.
- Outlier and dense areas **visible in t-SNE may not be them** in the n-D space.
- The meaningful similarity between n-D points can be **non-metric**.
- t-SNE similarity in 2-D **may not reflect n-D data structures** like in the Johnson-Lindenstrauss lemma above.
- This is the **fundamental deficiency of all n-D point-to-2-D point methods**.

- Therefore, we focus on **point-to-graph GLC methods**, which open a new opportunity to address this deficiency – **loss of n-D structure and properties**.
- However, we can see the statements that users **can easily identify outliers** in t-SNE. It will be valid only by showing that the n-D data similarity is **preserved** in 2-D.
- AtSNE algorithm [Cong Fu et al., 2019] is to resolve this t-SNE difficulty by capturing the global n-D data structure with 2-D anchor points (skeleton).
  - Again, it may not be possible (see the Johnson-Lindenstrauss lemma).

MNIST handwritten images are visualized as colored rectangles with their labels. “...one can easily identify which digit clusters are similar ... and which digit images are outliers (and thus confusing as another digit).”

Review of Visual Knowledge Discovery Approaches

1. Convert n-D data to 2-D data and then discover 2-D patterns in visualization as points.

2. Discover n-D patterns in n-D data then visualize them in 2-D as graphs.

3. Join 1 and 2: some patterns are discovered in (1) with controlled errors and some are discovered in (2).
1. Convert n-D data to 2-D data and then discover **2-D patterns** in visualization

**• Lossy approach**
- **Lossy** conversion to 2-D (dimension reduction, DR)
- **Point to point (n-D point to 2-D point)**
  - Visualization in 2-D
  - Interactive discovery of 2-D patterns in visualization

**• Lossless approach**
- **Lossless** conversion (visualization) to 2-D (n-D data fully restorable from its visualization) Point to graph
  - **(n-D point to graph in 2-D)**, e.g., \((5,4,0,6,5,10)\) to a graph
  - Interactive discovery of 2-D patterns on graphs in visualization
**Examples:** Example 1: linear discrimination of 4-D data, n-D point to graph

**Algorithm GLC-L**
- 4 coordinate lines: $X_1, X_2, X_3, X_4$ -- black lines (vectors) at different angles $Q_1$-$Q_4$
- 4-D point $(x_1, x_2, x_3, x_4) = (1, 0.8, 1.2, 1)$
  with these values shown as blue lines (vectors)
- Shifting and stacking blue lines
- Projecting the last point to U line
- Do the same for other 4-D points of blue class
- Do the same for 4-D points of red class
- Optimize angles $Q_1$-$Q_4$ to separate classes (yellow line)
Example 2: Visual linear discrimination of 9-D Wisconsin Breast Cancer data in GLC-L coordinates

- Critical in Medical diagnostics and many other fields:
  - Explanation of patterns and
  - Understanding them
- Lossless visual means
- Reversible/restorable

Only one malignant (red case) on the wrong side

Angles $Q_1$-$Q_9$ – contributions of attributes $X_1$-$X_9$

<table>
<thead>
<tr>
<th>Real Class</th>
<th>Predicted Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 424 20</td>
</tr>
<tr>
<td>2</td>
<td>1 238</td>
</tr>
</tbody>
</table>

Accuracy is: 96.9253%
Example 3: Visual discrimination of 9-D Wisconsin Breast Cancer data in Shifted Paired Coordinates (SPC)

Point a in $(X_1, X_2), (X_3, X_4), (X_5, X_6)$ as a sequence of pairs $(3, 2), (1, 4)$ and $(2, 6)$.

Point a in $(X_2, X_1), (X_3, X_5), (X_4, X_6)$ as a sequence of pairs $(2, 3), (1, 6)$ and $(2, 4)$.

A set of 688 Wisconsin Breast Cancer (WBC) data visualized in SPC as 2-D graphs of 10-D points with benign cases in red and malignant cases in blue.

WBC data in 4-D SPC as graphs in coordinates $(X_6, X_6)$ and $(X_8, X_7)$ that are used by Rule 1, i.e., WBC cases that go through $R_1$ and not go to $R_2$ and $R_3$ in these coordinates.

The WBC explainable classification **Rule** is

If $(x_8, x_9) \in R_1 \& (x_6, x_7) \notin R_2 \& (x_6, x_7) \notin R_3$

then $x \in$ class 1 (Red, Benign) else $x \in$ class 2 (Blue, Malignant)

This rule classifies all cases that are either in $R_2$ or in $R_3$ or not in $R_1$ as blue.

It has accuracy 93.60% $(425+219)/688$ on all 688 cases.

WBC cases covered by else in rule 2 (dominated by blue class).

Here malignant cases (blue cases)
Example 4: Avoiding Occlusion with Deep Learning on WBC data

1. Numeric 9-D Wisconsin Breast Cancer (WBC) Data
2. Visualized numeric data as downscaled 25x25 pixels images using GLC-L method
3. Deep Learning Convolutional Neural Network (CNN) on images (GLC-L visualization) as input
4. Classification accuracy 97.22% at the level and above published in literature
5. 10-fold cross validation
6. WBC data samples visualized in GLC-L for CNN model with the best accuracy.

Visualization optimization
EXAMPLE 5: INTERPRETABLE DCP ALGORITHM ON WBC DATA

1. Constructing dominance intervals for classes on each attribute
2. Combining dominance intervals in the voting methods
3. Learning parameters of dominance intervals and voting methods for prediction
4. Visualizing the dominance structure
5. Explaining the prediction

Average accuracy of 10-fold cross validation is 97.01%
Improved algorithm 99.3%
General Line Coordinates (GLC) to convert n-D points to graphs in 2-D
General Line Coordinates (GLC) 

Descartes lay in bed and invented the method of co-ordinate geometry. 

Alfred North Whitehead

7-D point $D=(5,2,5,1,7,4,1)$ in Radial Coordinates.

7-D point $D=(5,2,5,1,7,4,1)$ in Parallel Coordinates

(a) 7-D point $D$ in General Line Coordinates with straight lines.

(b) 7-D point $D$ in General Line Coordinates with curvilinear lines.

(c) 7-D points $F-J$ in General Line Coordinates that form a simple single straight line.

(d) 7-D points $F-J$ in General Line Coordinates that do not form a simple single straight line.

7-D points in General Line Coordinates with different directions of coordinates $X_1, X_2, \ldots, X_7$ in comparison with Parallel Coordinates.
General Line Coordinates (GLC)

n-Gon (rectangular) coordinates with 6-D point (0.5, 0.6, 0.9, 0.7, 0.7, 0.1).

3-D point A=(0.3, 0.7, 0.4) in 3-Gon (triangular) coordinates and in radial coordinates.

Figure 2.5. Examples of circular coordinates in comparison with parallel coordinates.

Figure 2.6 Example of weekly stock data in n-Gon (pentagon) coordinates.
General Line Coordinates (GLC): 2-D visualization

2-D Line Coordinates.

<table>
<thead>
<tr>
<th>Type</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-D General Line Coordinates (GLC)</td>
<td>Drawing n coordinate axes in 2-D in variety of ways: curved, parallel, unparalleled, collocated, disconnected, etc.</td>
</tr>
<tr>
<td>Collocated Paired Coordinates (CPC)</td>
<td>Splitting an n-D point ( x ) into pairs of its coordinates ( (x_1,x_2),\ldots,(x_{n-1},x_n) ); drawing each pair as a 2-D point in the collocated axes; and linking these points to form a directed graph. For odd ( n ) coordinate ( X_n ) is repeated to make ( n ) even.</td>
</tr>
<tr>
<td>Basic Shifted Paired Coordinates (SPC)</td>
<td>Drawing each next pair in the shifted coordinate system by adding ((1,1)) to the second pair, ((2,2)) to the third pair, ((i-1,i-1)) to the (i)-th pair, and so on. More generally, shifts can be a function of some parameters.</td>
</tr>
<tr>
<td>2-D Anchored Paired Coordinates (APC)</td>
<td>Drawing each next pair in the shifted coordinate system, i.e., coordinates shifted to the location of a given pair (anchor), e.g., the first pair of a given n-D point. Pairs are shown relative to the anchor easing the comparison with it.</td>
</tr>
<tr>
<td>2-D Partially Collocated Coordinates (PCC)</td>
<td>Drawing some coordinate axes in 2D collocated and some coordinates not collocated.</td>
</tr>
<tr>
<td>In-Line Coordinates (ILC)</td>
<td>Drawing all coordinate axes in 2D located one after another on a single straight line.</td>
</tr>
<tr>
<td>Circular and n-Gon coordinates</td>
<td>Drawing all coordinate axes in 2D located on a circle or an n-Gon one after another.</td>
</tr>
<tr>
<td>Elliptic coordinates</td>
<td>Drawing all coordinate axes in 2D located on ellipses.</td>
</tr>
<tr>
<td>GLC for linear functions (GLC-L)</td>
<td>Drawing all coordinates in 2D dynamically depending on coefficients of the linear function and value of n attributes.</td>
</tr>
<tr>
<td>Paired Crown Coordinates (PWC)</td>
<td>Drawing odd coordinates collocated on the closed convex hull in 2-D and even coordinates orthogonal to them as a function of the odd coordinate.</td>
</tr>
</tbody>
</table>
# General Line Coordinates (GLC): 3-D visualization

3-D Line Coordinates.

<table>
<thead>
<tr>
<th>Type</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-D General Line Coordinates (GLC)</td>
<td>Drawing $n$ coordinate axes in 3-D in variety of ways: curved, parallel, unpara-</td>
</tr>
<tr>
<td></td>
<td>lleled, collocated, disconnected, etc.</td>
</tr>
<tr>
<td>Collocated Tripled Coordinates (CTC)</td>
<td>Splitting $n$ coordinates into triples and representing each triple as 3-D point in the same three axes; and linking these points to form a directed graph. If $n \mod 3$ is not 0 then repeat the last coordinate $X_n$ one or two times to make it 0.</td>
</tr>
<tr>
<td>Basic Shifted Tripled Coordinates (STC)</td>
<td>Drawing each next triple in the shifted coordinate system by adding (1,1,1) to the second triple, (2,2,2) to the third triple $(i-1, i-1,i-1)$ to the $i$-th triple, and so on. More generally, shifts can be a function of some parameters.</td>
</tr>
<tr>
<td>Anchored Tripled Coordinates (ATC) in 3-D</td>
<td>Drawing each next triple in the shifted coordinate system, i.e., coordinates shifted to the location of the given triple of (anchor), e.g., the first triple of a given n-D point. Triple are shown relative to the anchor easing the comparison with it.</td>
</tr>
<tr>
<td>3-D Partially Collocated Coordinates (PCC)</td>
<td>Drawing some coordinate axes in 3-D collocated and some coordinates not collocated.</td>
</tr>
<tr>
<td>3-D In-Line Coordinates (ILC)</td>
<td>Drawing all coordinate axes in 3D located one after another on a single straight line.</td>
</tr>
<tr>
<td>In-Plane Coordinates (IPC)</td>
<td>Drawing all coordinate axes in 3D located on a single plane (2-D GLC embedded to 3-D).</td>
</tr>
<tr>
<td>Spherical and polyhedron coordinates</td>
<td>Drawing all coordinate axes in 3D located on a sphere or a polyhedron.</td>
</tr>
<tr>
<td>Ellipsoidal coordinates</td>
<td>Drawing all coordinate axes in 3D located on ellipsoids.</td>
</tr>
<tr>
<td>GLC for linear functions (GLC-L)</td>
<td>Drawing all coordinates in 3D dynamically depending on coefficients of the linear function and value of $n$ attributes.</td>
</tr>
<tr>
<td>Paired Crown Coordinates (PWC)</td>
<td>Drawing odd coordinates collocated on the closed convex hull in 3-D and even coordinates orthogonal to them as a function of the odd coordinate value.</td>
</tr>
</tbody>
</table>
Reversible Lossless Paired Coordinates

(a) Collocated Paired Coordinates.
(b) Shifted Paired Coordinates.

6-D point \((5, 4, 0, 6, 4, 10)\) in Paired Coordinates.

State vector \(x = (x, y, x', y', x'', y'') = (0.2, 0.4, 0.1, 0.6, 0.4, 0.8)\) in Collocated Paired and Parallel Coordinates.

Anchored Paired Coordinates with numbered arrows.
Reversible lossless Paired Coordinates

6-D point as a closed contour in 2-D where a 6-D point $x=(1,1,2,2,1,1)$ is forming a tringle from the edges of the graph in Paired Radial Coordinates with non-orthogonal Cartesian mapping.

6-D point $(1,1,1,1,1,1)$ in two $X_1$-$X_6$ coordinate systems (left – in Radial Collocated Coordinates, right – in Cartesian Collocated Coordinates).

n-D points as closed contours in 2-D: (a) 16-D point $(1,1,2,1,1,2,1,1,2,1,1,2,1,1,2,2)$ in Partially Collocated Radial Coordinates with Cartesian encoding, (b) CPC star of a 192-D point in Polar encoding, (c) the same 192-D point as a traditional star in Polar encoding.

4-D point $P=(0.3,0.5,0.5,0.2)$ in 4-D Elliptic Paired Coordinates, EPC-H as a green arrow. Red marks separate coordinates in the Coordinate ellipse.

4-D point $P=(0.3,0.5,0.5,0.2)$ in Radial Coordinates.
Reversible lossless Paired Coordinates

Partially Collocated Orthogonal (Ortho) Coordinates.

Partially Collocated Ortho and non-Ortho Coordinates.

Collocated Paired non-Ortho Coordinates.

In Figure (a) below, the value \( w_5 \) (labeled with 1) is a starting point of the normal to the square that forms the coordinate \( X_6 \). The length of this norm is the value of \( w_6 \).

(a) Mapping odd coordinates to a square.  
(b) Mapping odd coordinates to a circle.

34-D point from figure 2.32.

Closed Paired Crown Coordinates with odd coordinates mapped to the crown and even coordinates mapped dynamically to the normals to the crown.
Graph construction algorithms in GLC

Six coordinates and six vectors that represent a 6-D data point (0.75,0.5,0.7,0.6,0.7, 0.3) in GLC-SC1.

6-D data point (0.75,0.5,0.7,0.6,0.7, 0.3) in GLC-PC.

6-D data point (0.75,0.5,0.7,0.6,0.7, 0.3) in GLC-SC2.

6-D data point (0.75,0.5,0.7,0.6,0.7, 0.3) in GLC-CC1.

6-D data point (0.75,0.5,0.7,0.6,0.7, 0.3) in GLC-CC2.
Math, theory and pattern simplification methodology: Statements

- **Statement 1.** Parallel Coordinates, CPC and SPC preserve $L^p$ distances for $p=1$ and $p=2$, $D(x,y) = D^*(x^*,y^*)$.

- **Statement 2 (n points lossless representation).** If all coordinates $X_i$ do not overlap then GLC-PC algorithm provides bijective 1:1 mapping of any n-D point $x$ to 2-D directed graph $x^*$.

- **Statement 3 (n points lossless representation).** If all coordinates $X_i$ do not overlap then GLC-PC and GLC-SC1 algorithms provide bijective 1:1 mapping of any n-D point $x$ to 2-D directed graph $x^*$.

- **Statement 4 (n/2 points lossless representation).** If coordinates $X_i$, and $X_{i+1}$ are not collinear in each pair $(X_i, X_{i+1})$ then GLC-CC1 algorithm provides bijective 1:1 mapping of any n-D point $x$ to 2-D directed graph $x^*$ with $\lceil n/2 \rceil$ nodes and $\lceil n/2 \rceil - 1$ edges.

- **Statement 5 (n/2 points lossless representation).** If coordinates $X_i$, and $X_{i+1}$ are not collinear in each pair $(X_i, X_{i+1})$ then GLC-CC2 algorithm provides bijective 1:1 mapping of any n-D point $x$ to 2-D directed graph $x^*$ with $\lceil n/2 \rceil$ nodes and $\lceil n/2 \rceil - 1$ edges.
Math, theory and pattern simplification methodology:

**Statements**

- **Statement 6** *(n points lossless representation)*. If all coordinates $X_i$ do not overlap then GLC-SC2 algorithm provides bijective 1:1 mapping of any n-D point $x$ to 2-D directed graph $x^*$.  

- **Statement 7**. GLC-CC1 preserves $L^p$ distances for $p=1$, $D(x,y) = D^*(x^*,y^*)$.  

- **Statement 8**. In the coordinate system $X_1,X_2,...,X_n$ constructed by the Single Point algorithm with the given base n-D point $x=(x_1, x_2,...,x_n)$ and the anchor 2-D point $A$, the n-D point $x$ is mapped one-to-one to a single 2-D point $A$ by GLC-CC algorithm.  

- **Statement 9** *(locality statement)*. All graphs that represent nodes $N$ of n-D hypercube $H$ are within square $S$.
Adjustable GLCs for decreasing occlusion and pattern simplification

Non-preattentive vs. preattentive visualizations (linearized patterns): 6-D point $A = (3, 6, 4, 8, 2, 9)$ in blue, and 6-D point $B = (3.5, 6.8, 4.8, 8.5, 2.8, 9.8)$ in orange in Traditional, Shifted Parallel Coordinates, and GLC.

Simplicity is the ultimate sophistication.
Leonardo da Vinci
Case Studies: World Hunger data

4-D data: representation of prevalence of undernourished in the population (%) in Collocated Paired Coordinates

4-D data: representation of prevalence of undernourished in the population (%) in traditional time series (equivalent to Parallel Coordinates for time series)

The Global Hunger Index (GHI) for each country measures as,

\[
GHI = \frac{\text{UNN} + \text{UW5} + \text{MR5}}{3},
\]

where UNN is the proportion of the population that is Undernourished (in %), UW5 is the prevalence of Underweight in children under age of five (in %), and MR5 is the Mortality rate of Children under age five (in %).
Challenger Space Shuttle data

The Challenger O-rings data: (1) temporal order of flight, (2) number of O-rings at risk on a given flight, (3) number of O-rings experiencing thermal distress, (5) launch temperature (degrees F), and (5) leak-check pressure (psi). These data have been normalized to be in the [0,1].

Challenger USA Space Shuttle normalized O-Ring dataset in the Collocated Paired Coordinates (CPC). X, Y coordinates are normalized values of attributes. 6 O-rings at risk are normalized as 0.

CPC shows three distinct flights #2, #14, and #2 with arrows going down and horizontally. These flights had maximum value of O-rings at risk. Thus CPC visually show a distinct pattern with a meaningful interpretation. The well-known case is #14, which is the lowest temperature (53°F) from the previous 23 Space Shuttle launches. It stands out in the CPC.

The case #2 also experienced thermal distress for one O-ring at much higher temperature of 70°F and lower leak-check pressure (50 psi). This is even more outstanding from others with the vector directed down.

The case #14 is directed horizontally. All other cases excluding case #21 are directed up.
Case Studies: Health Monitoring with PC and CPC

The colors show the progress to the goal.

- **Dark green dot** – goal.
- **Yellow and light green** -- closer to the goal point.
- **Red arrow** – initial health status.

- **Experiments** -- people quickly grasp how to use this health monitor.
- **This health monitor is expandable.**
- **Two more indicators is another pair of shifted Cartesian Coordinates.**
  - *The goal is the same dark green 2-D dot*
  - *Each graph has two connected arrows.*
- **Graphs closer to the goal are smaller.**

a) PSPC: The green dot is the desired goal state, the red arrow is the initial state, the orange arrow is the health state at the next monitoring time, and the light green arrow is the current health state of the person.

4-D Health monitoring visualization in PSPC (a) and Parallel Coordinates (b) with parameters: systolic blood pressure, diastolic blood pressure, pulse, and total cholesterol at four time moments.
Case studies: Knowledge Discovery and Machine Learning for Investment Strategy with CPC

- The CPC visualization shows arrows in $(V,Y)$ space of volume $V_r$ and relative main outcome variable $Y_r$.
- This is a part of the data shown as traditional time series with time axis.
- CPC has no time axis. The arrow direction shows time.
- The arrow beginning is the point in the space $(V_r,Y_r)$, and its head is the next time point in the collocated space $(V_{r+1},Y_{r+1})$.
- CPC gives the inspiration idea for building a trading strategy in contrast with time series figure without it.
  - It allows finding the areas with clusters of two kinds of arrows.
  - The arrows for the long positions are green arrows.
  - The arrows for the short positions, are red.
  - Along the $Y_r$ axis we can observe a type of change in $Y$ in the current candle. If $Y_{r+1}>Y_r$ then $Y_{r+1}>Y_r$; the right decision in i-point is a long position opening. Otherwise, it is a short position.
  - Next, CPC shows the effectiveness a decision in the positions.
  - The very horizontal arrows indicate small profit.
  - A more vertical arrows indicates the larger profit.
- In comparison with traditional time series, the CPC brings the additional knowledge about the potential of profit in selected area of parameters in $(V,Y)$ space.
The zoomed cubes with the best asymmetry from Figure 8.16. The upper cube with green circles is selected for long positions lower cube with red pins in 3-D space: two cubes found in \((Y_r, dMA_r, V_r)\) space with the maximum asymmetry between long and short positions.

Two determined cubes in \(Y_r-dMA_r-V_r\) space with the maximum asymmetry between long and short positions for the new grid resolution.
(a) Bars in 3D space after learning period which represent preferences to open long (green) or short (red) position.

(b) Cumulative profits for different values of parameters \( (f,s,k_b) \).

Bars in 3D space after learning period which represent preferences to open long (green) or short (red) position and cumulative profits for different values of parameters \( (f,s,k_b) \).

- The thicker line is the best one relative to the value of a linear combination of the cumulative profits for learning period and Calmar ratio in the same learning period.
- The main idea in constructing the criterion \( C \) is to balance risk and profit contributions with different pairs of weights \( (w_c, w_p) \).

- Criterion to select the promising curve:
  \[
  C = w_c \cdot Calmar_{learn} + w_p \cdot profit_{learn}
  \]
  where \( w_c \) is a weight of Calmar component (in these experiments \( w_c = 0.3 \)); \( w_p \) is a weight of profit component (in these experiments \( w_p = 100 \) for 500 hours of learning period); \( Calmar_{learn} \) is the Calmar ratio at the end of learning period; and \( profit_{learn} \) is a cumulative profit at the end of the learning period measured as a change in EURUSD rate.

- Calmar ratio, \( CR = \text{(average of return over time } \Delta t)/\text{max drawdown over time } \Delta t \), which shows the quality of trading. “Calmar ratio of more than 5 is considered excellent, a ratio of 2 – 5 is very good, and 1 – 2 is just good” [Main, 2015].

- The curve with a significant profit and a small variance (captured by larger Calmar ratio for evaluating risk) is used in criterion \( C \) as a measure of success.
Visual Text Mining: Discovery of Incongruity in Humor Modeling

All intellectual labor is inherently humorous.
George Bernard Shaw

Incongruity process for model $M$.

<table>
<thead>
<tr>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>Agent $G$ (human or a software agent) reads the first part of the text $P_1$ and concludes (at a surface level) that $P_1$ is a usual text.</td>
</tr>
<tr>
<td>$t_2$</td>
<td>Agent $G$ reads the second part of the text $P_2$ and concludes that (at a surface level) that $P_2$ is a usual text.</td>
</tr>
<tr>
<td>$t_3$</td>
<td>Agent $G$ starts to analyze a relation between $P_1$ and $P_2$ (at a deeper semantic level). Agent $G$ retrieves semantic features (words, phrases) $F(P_1)$ of $P_1$.</td>
</tr>
<tr>
<td>$t_4$</td>
<td>Agent $G$ retrieves semantic features (words, phrases) $F(P_2)$ of $P_2$.</td>
</tr>
<tr>
<td>$t_5$</td>
<td>Agent $G$ compares (correlates) features $F(P_1)$ with $P_2$ and features $F(P_2)$ with $P_1$ finding significant differences in meaning (incongruity).</td>
</tr>
<tr>
<td>$t_6$</td>
<td>Agent $G$ reevaluate usuality of $P_2$ taking into account these correlations and concludes that $P_2$ is unusual.</td>
</tr>
</tbody>
</table>

Confusion matrix for visual classification rule $R_1$

<table>
<thead>
<tr>
<th></th>
<th>Predicted joke</th>
<th>Predicted non-joke</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual jokes 17</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Actual non-jokes 17</td>
<td>2</td>
<td>15</td>
</tr>
</tbody>
</table>

We record this visual discovery as a rule $R_2$:

$R_2$: If $z_3 < z_4$ then $z$ is a joke, else $z$ is a non-joke.

The resulting C4.5 rule $R_3$ is

If $z_3 < z_4 < 0.0075$ then $z$ is a joke, else $z$ is a non-joke.

R4: If $(w \geq (1,0,1,1)) \lor (w \geq (0,1,0,1))$ then $w$ is joke else $w$ is a non-joke.
Case study: Recognition of digits
Dimension reduction

Experiments with 900 samples of MNIST dataset for digits 0 and 1. Results of the automatic dimension reduction displaying 249 dimensions with 235 dimensions removed with the accuracy dropped by 0.28%.
Case study: Recognition of digits

Dimension reduction

Lossy but controlled Dimension Reduction with NN autoencoder

- Input layer = Output layer
- 484 nodes each
- Hidden layer has 24 nodes

A subset of the testing set between 50 to 100 samples for each digit
- 24-dimensional data with 10 different classes
- 24/784 is 3% after preprocessing and decoding

Cropped input Images 22x22 = 484 pixels
cropped edges from 784 pixels

Decoded images with hidden Layer: 24 nodes
Case study: Digit recognition, Dimension reduction

Encoded digit 0 and digit 1 on and GLC-L, using 24 dimensions found by the Autoencoder among 484 dimensions. Results for the best linear discriminant function of the first run of 20 epochs.

Encoded digit 1 in Parallel Coordinates using 24 dimensions found by the Autoencoder among 484 dimensions. Each vertical line is one of the 24 features scaled in the [0,35] interval. Digit 1 is visualized on the parallel coordinates.

Encoded digit 0 in Parallel Coordinates using 24 dimensions found by the Autoencoder among 484 dimensions. Each vertical line is one of the 24 features scaled in the [0,35] interval. Digit 0 is visualized in Parallel Coordinates.

Encoded digit 0 and digit 1 on GLC-L, using 24 dimensions found by the Autoencoder among 484 dimensions. Another run of 20 epochs, best linear discriminant function from this run. Accuracy drops 1%.
How to classify 0 and 1 using these visualizations?
- Select discriminating features.
- Black circles show the differences between 0 and 1 digits in these 24 coordinates.
- Design rules and explain them, e.g.,
  if \(x_2 = 0\) then 0; if \(x_7 = 0\) then 0.
  If \(x_{24}\) is in green oval then 0.
- Remove cases that satisfy these simple rules
- Search rules form remaining cases

How to interpret these 24 features?
- Tracing these 24 features to find their origin in the image.

Comparing encoded digit 0 and digit 1 on the parallel coordinates, using 24 dimensions found by the Autoencoder among 484 dimensions. Each vertical line is one of the 24 dimensions.
Collaborative visual discovery

Data and Task Split-based Collaborative Visualization framework.

Collaboration diagram with data example 1.

Experiments: success in dimension $n>100$
Current Software
Case study: Recognition of digits

MNIST --benchmark dataset for hand written digit recognition 60,000 training samples and 10,000 testing samples

MNIST subset for digits 0 and 1 data set showing training dataset (70% of the whole dataset) when trained on the training dataset to find the coefficients.

MNIST subset for digits 0 and 1 dataset showing validation dataset (30% of the whole dataset) when trained on the training dataset to find the coefficients. Projecting validation set.
Summary on General Line Coordinates

- The examples and cases studies show that hybrid methods with General Line Coordinates are capable
  - **visualizing** data of multiple dimensions from 4-D to 484-D without loss of information and
  - **discovering** patterns by combining humans perceptual capabilities and Machine Learning/Data Mining algorithms for classification such high-dimensional data.
- This Hybrid technique can be developed further in multiple ways to deal with different new challenging data science tasks.
Approaches to explain deep and other ML models by visual means

- Explaining ML models including deep learning models by visual means
  - activation and weight visualization,
  - heat-map-based methods,
  - dependency analysis,
  - monotonicity approach – monotone Boolean functions and chains;
  - decision tree visualization, and others.
Tools for explaining deep learning models

- Tools that **visualize activations** generated on every layer of a trained convolutional net during image/video recognition.
  - Benefits: builds intuitions how the network work (**algorithm tracing**).

- Tool that **visualize features** at every layer of a network discovered by optimization in image space.
  - Challenge --- blurred and less recognizable images of features.
  - Approach: new **regularization methods** for optimization to generate clearer, more interpretable visualizations.

- A new term -- **deep visualization**
CIFAR-10 classification benchmark problem is to classify RGB 32x32 pixel images across 10 categories.

- Tensor flow CIFAR-10 CNN multi-layer network with
  - alternating convolutions and nonlinearities followed by fully connected layers and softmax classifier.
  - Peak accuracy ~ 86%
  - few hours of training on a GPU.
  - ~ 1M learnable parameters
  - 19.5M multiply-add operations to compute inference on a single image.

Discovering human explainable visualization of what a deep neural network has learned

- Comparison of three heatmaps for digit ‘3’.
- Left: *The randomly generated heatmap* – no interpretable information.
- Middle: *The segmentation heatmap* – shows the whole digit **without relevant parts**, say, for distinguishing ‘3’ from ‘8’ or ‘9’.
- Right: *A relevance heatmap* – shows parts of the image **used by the classifier**. 
  - Reflects human intuition on differences between ‘3’, ‘8’ and ‘9’ and other digits.

Comparison of the three heatmap computations

**Left:**
- **Sensitivity heatmaps** (local explanations) – measure change of the class when specific pixels are changed based on partial derivatives.
- Applicable to architectures with differentiable units.

**Middle:**
- **Deconvolution method** ("autoencoder") – applies a convolutional network $g$ to the output of another convolutional network $f$. Network $g$ “undoes” $f$.
- Challenge: Unclear relation between heatmap scores and the classification output $f(x)$.

**Right:**
- **Layer-wise Relevance Propagation** (LRP) – exactly decomposes the classification output $f(x)$ into pixel relevancies by observing the layer-wise evidence for class preservation (conservation principle).
- Applicable to generic architectures (including with non-continuous units) – does not use gradients.
- Globally explains the classification decision and heatmap scores
- Have a clear interpretation as evidence for or against a class.

---

Multifaceted Feature Visualization

Based on “fooling” images, scientists previously concluded that DNNs trained with supervised learning ignore an object’s global structure, and instead only learn a few, discriminative features per class (e.g. color or texture) (Nguyen et al., 2015).

- Multifaceted Feature Visualization algorithm:
  - shows the multiple feature facets each neuron detects,
  - reduces optimization’s tendency to produce repeated object fragments
  - tends to produce one central object.

- Benefits
  - more comprehensive understanding of each neuron’s function.
  - increases understanding of DNN,
  - opportunity to create more powerful DNN algorithms.

Learning deep features for scene recognition using places database

- **Challenge:**
  - performance at **scene recognition** is lower than for object recognition.

- **Reasons:**
  - Current deep features trained from **ImageNet** are not competitive enough for such tasks.

- **Approach:**
  - new scene-centric database called **Places** with over **7 million labeled pictures of scenes**.
  - New methods to compare the **density and diversity** of image datasets
  - **CNN** to learn deep features for scene recognition
  - **Visualization** of the CNN layers’ responses to show differences in the internal representations of object-centric and scene-centric networks.

Heat map visualization in identification of diabetic retinopathy using deep learning

- A common method of combining results of Deep Learning (DL) from images with visualization is:
  - discovering classification model for images using a DL algorithm,
  - identifying informative deep features, and
  - visualizing identified deep features on the original image.

- The methods of visualization of these features range from
  - outlining the area of deep features to
  - overlaying the heat map in these areas (B in the image).

- The last method was used for retinal images in diabetic retinopathy diagnostics.

Unsupervised feature learning for audio classification using Convolutional Deep Belief Network

- **Computational Approach**
  - Convert time-domain signals into spectrograms.
  - Extract spectrogram with overlaps from each TIMIT training data case.
  - **Reduce dimension** of the spectrogram by PCA to 80 principal components.
  - **Training** first-layer cases. Training second-layer cases using first-layer activations as input.

- **Visualization Approach**
  - Finding what the network “learns”.
  - Visualizing the first layer bases
  - Visualizing second-layer as a weighted linear combination of the first-layer bases.

A spectrogram is a **heatmap** image with axes for time and frequency and intensity or color of a point for amplitude.

**TIMIT** — recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences.

Heat maps visualization and explanation of deep learning for pulmonary tuberculosis

Chest radiograph with pathologically proven active TB.

The same radiograph with a heat map overlay of a strongest activations from the 5th convolutional layer from GoogLeNet-TA classifier.

The red and light blue regions in the upper lobes -- areas activated by the deep neural network. (areas where the disease is present)
The dark purple background -- areas that are not activated.

Nonlinear feature space dimension reduction in breast computer-aided diagnosis (CADx)

Goals:
- Enhancing breast CADx with unsupervised dimension reduction (DR)
- Representation of computer-extracted breast lesion feature spaces for
  - 1126 ultrasound cases,
  - 356 MRI cases,
  - 245 mammography 245 cases.

Evaluation criteria:
- Performance of classifiers with DR relative to malignancy classification
- Visual inspection of sparseness of 2-D and 3-D mappings.

Results:
- Slightly higher performance of 4-D t-SNE mapping (from the original 81-D feature space) relative to 13 and 4 selected features for other methods.

Lossy visualizations for ML models

- 2D and 3D visualizations of unsupervised reduced dimension representations of 81-D breast lesion ultrasound feature data green – Benign lesions, Red - Malignant, Yellow - Benign-cystic.

- Visualization of linear reduction using
  - (a) PCA, first two principal components
  - (b) first three principal components, 3D PCA.
  - (c) 2D and (d) 3D visualization of the nonlinear reduction mapping using t-SNE

- Discussion:
  - Methods like T-SNE, PCA and others do not preserve all information of initial features (they are lossy visualizations of n-D data)
  - They convert 81 interpretable features to 2-3 artificial features that have no direct interpretation
  - General Line Coordinates is an alternative that preserves all n-D information when occlusion/clutter in visualization is suppressed that was successfully done in [Kovalerchuk et al, 2014-2019]


T-SNE visualization for explainable AI in intrusion detection systems: an adversarial approach

Goal:
- Increase understanding of “black-box” DNN models.

Approach:
- Explain miss-classifications made by Intrusion Detection Systems
- Find minimum modifications (of the input features) to correct misclassification by using adversarial machine learning
- Make intuitive visualization of magnitude of max modifications as most explanatory features for the misclassification.

Tests:
- on the NSL-KDD99 benchmark data using Linear and Multilayer perceptrons that match expert knowledge.

The advantages:
- Applicable to any classifier with defined gradients.
- Does not require modification of the classifier model.

Discussion:
- Lossy T-SNE does not preserve all information of modified values of initial features.
- T-SNE converts interpretable features to 2-3 artificial features that have no direct interpretation
- General Line Coordinates is an alternative to T-SNE that preserve all n-D information, but potentially suffer more from occlusion/clutter in visualization.

Approaches to discover analytical ML models assisted by visual means

- Discovering visual ML models assisted by analytical ML algorithms, such as
  - propositional and first order rules,
  - random forests,
  - CNN,
  - decision trees,
  - optimization based on genetic and other algorithms
Quality-based guidance for exploratory dimensionality reduction

Problem:
- Difficulty or inability to represent effectively high-dimensional data with hundreds of variables by visualization methods

Known solutions:
- Employing dimensionality reduction (DR) prior to visualization.

Challenge:
- DR can throw a baby with bathwater – loss of information of full high-dimensional data.

Approach: interactive environment to understand n-D data first with:
- Discovering structure of full n-dimensional data.
- Identifying importance and interestingness of structures and variables.
- Using several metrics of ‘interestingness’ of variables as new features.
- Using PCA to combine metrics and get new “principal” metrics.

Use case:
- DNA sequence-based study of bacterial populations.

Workflow

The primary window of the interactive environment displaying a bacterial population data set, which includes 184 variables. The quality metric profiles of the variables are displayed in the Ranking and quality view (top left view) and in the Glyph view (top right view). The high-dimensional data set is displayed in the bottom view.

Quality-based guidance for exploratory dimensionality reduction

Top – visualizations of quality metrics of variables using parallel coordinates and glyphs.
In **parallel coordinates**, axes are quality metrics, and polylines are the variables.
In the **glyph plot**, axes are first two “principal metrics”.
Glyphs are located at the points with values of these metrics for each variable. Glyph is a set of squares: each square represents one metric opacity of the square represents the metric value.
The fill color of the square is the same as the axis of the parallel coordinates.
The border color of the glyph is the color of polylines in parallel coordinates.

Highest ranked operational taxonomic units (OTUs) in a scatter plot matrix, where positive and negative correlations are represented by blue and red cells.

The variable merging window, including a list of suggestions for variable groups to merge.
An Interactive Large-Scale Graph Mining and Visualization

Problem:
- Exploring efficiently a large graph with several millions or billions of nodes and edges, such as a social network?

Solution:
- Perseus, an integrative system for analysis of large graphs

Approach:
- summarization of graph properties and structures,
- guiding attention to outliers,
- exploration of normal and anomalous node behaviors,
- automatic extraction of graph invariants (e.g., degree, PageRank, real eigenvectors)
- scalable online batch processing on Hadoop;
- visualization of 1-D and 2-D distributions of invariants
- Visualization of a subgraph of the selected node and its neighbors, by incrementally revealing its neighbors.

Use Cases: multi-million-edge social networks
- Wikipedia vote network,
- friendship/foeship network in Slashdot, and
- trust network based on the consumer review website Epinions.com.

Contrastive and Visual Topic Modeling for Comparing Documents

- **Problem:**
  - How to express similarities/differences of ‘labeled’ document collections and visualize them efficiently?

- **Approach**
  - Learn hidden topics and embeddings for the documents, topics and labels for visualization.
  - Extract hidden discriminative and common topics across labeled documents.
  - Visualize all documents, labels, and the extracted topics, where proximity in the coordinate space is reflective of proximity in semantic space;
  - extract topics and visual coordinates simultaneously under a joint model.
  - Probabilistic approach to create a visualization space that differs from t-SNE.

- **Results:**
  - Outperforms both unsupervised and supervised state-of-the-art topic models in contrastive power, semantic coherence and visual effectiveness on real world data.

ContraVis: (supervised joint method) finds latent discriminative topics and gives visual embedding of documents and topics. Two discriminative topics per label shown in wordclouds (w/ respective colors) + one common topic (black);

Monotone Boolean Function visualization vs. Parallel Coordinates for binary data

(c) cancer visualization in centered chain order

(f) biopsy visualization in centered chain order

(j) Highly overlapped parallel coordinate visualization of the same data (yellow - benign, red –malignant)

(k) Types of source X-ray mammography images used producing Boolean vectors

Boolean Data and Hansel Chains

- Boolean Data are data composed of ‘0’ and ‘1’
- Hansel Chains browse without the binary cube or hyper cube without overlapping

Traditional graph representation uses space inefficiently (unused space)
the concept of the monotone Boolean function from discrete mathematics [Korshunov, 2003, Keller, Pilpel, 2009]. Let $E^n = \{0,1\}^n$ be a binary $n$-dimensional cube then vector $y = (y_1, y_2, \ldots, y_n)$ is no greater than vector $x = (x_1, x_2, \ldots, x_n)$ from $E^n$ if for every $i$ $x_i \geq y_i$, i.e.,

$$x \geq y \iff \forall i \ x_i \geq y_i$$

In other words, vectors $x$ and $y$ are ordered. In general relation $\geq$ for Boolean vectors in $E^n$ is a partial order that makes $E^n$ a lattice with a max element $(1,1,\ldots,1)$ and min element $(0,0,\ldots,0)$.

Boolean function $f: E^n \rightarrow E$ is called a monotone Boolean function if

$$\forall x \geq y \Rightarrow f(x) \geq f(y).$$

This monotonicity property implies two expansion properties for function $f$:

$$x \geq y \ & f(y) = 1 \Rightarrow f(x) = 1, \quad x \geq y \ & f(x) = 0 \Rightarrow f(y) = 0,$$
A Basic Example

- MDF without data

P₁ process numeric based vector location

Data with level/norm = 10

Data with level/norm = 5 (n/2)

Data with level/norm = 0
48-D and 96-D data in CPC-Stars, Radial Stars and Parallel Coordinates

Figure 6.3. Examples of corresponding figures: stars (row 1) and PCs lines (row 2) for five 48-D points from two tubes with $m = 5\%$. Row 3 and 4 are the same for dimension $n=96$.

Figure 6.4. Two stars with identical shape fragments on intervals $[a,b]$ and $[d,c]$ of coordinates.

Figure 6.5. Samples of some class features on Stars for $n=48$.

Figure 6.6. Samples of some class features on PCs for $n=48$. 
Human abilities to discover high-dimensional patterns

Figure 6.7. Twenty 160-D points of 2 classes represented in star CPC with noise 10% of max value of normalized coordinates (max=1) and with standard deviation 20% of each normalized coordinate.

Figure 6.8. Twenty 160-D points of 2 classes represented in Radial Coordinates with noise 10% of max value of normalized coordinates (max=1) and with standard deviation 20% of each normalized coordinate.

Figure 6.9. Twenty 160-D points of 2 classes represented in Parallel Coordinates with noise 10% of max value of normalized coordinates (max=1) and with standard deviation 20% of each normalized coordinate.

Figure 6.10. Samples of 100-D data in Star CPC used to make participants familiar with the task.
Human abilities to discover patterns in 170-D data

- The experiment with \( n = 192 \) and a high level of noise (30\%) points out on the likely upper bound of human classification of \( n \)-D data using the Radial Coordinates for data modeled as linear hyper-tubes.

- The experiment with \( n = 160 \) shows that the upper bound for human classification on such \( n \)-D data is no less than \( n = 160 \) dimensions with up to 20\% noise.

- Thus the expected classifiable dimensions are in \([160, 192]\) dimensions interval for the Radial Coordinates.

- Due to advantages of Star CPC over Radial Coordinates, these limits must be higher for Star CPC and lower for Parallel Coordinates due to higher occlusion in PC.

- More exact limits are the subject of the future experiments. About 70 respondents participated in the experiment with 160-D, therefore it seems that 160 dimensions can be viewed as a quite firm bound.

- In contrast, the question that 192-D is the max of the upper limit for Star CPC may need additional studies.

- Thus, so far the indications are that the upper limit for Star CPC is above \( n = 192 \) and it needs to be found in future experiments for linear hyper-tubes. Finding bounds for linear hyper-tubes most likely will be also limits for non-linear hyper-tubes due to their higher complexity.

Figure 6.17. Traditional 170-D stars: class “musk” (first row) and class “non-musk chemicals” (second row). CPC 170-D stars from the same dataset: class “musk” (third row) and class “non-musk chemicals” (forth row).

Figure 6.18. Nine 170-dimensional points of two classes in Parallel Coordinates.
Conclusion

■ The tutorial covered five complementary approaches:
  1. to visualize ML models produced by analytical ML methods,
  2. to discover ML models by visual means,
  3. to explain deep and other ML models by visual means,
  4. to discover visual ML models assisted by analytical ML algorithms,
  5. to discover analytical ML models assisted by visual means.

■ All of them benefit data science now and will continue in the future.
New tasks for visual knowledge discovery

- Expansion to new applications in finance, medicine, NLP, image processing and others
- Deepen links with Deep Learning methods
- Expansion of hybrid approach to prevent overgeneralization and overfitting of predictive models by using visual means.
- Multiple “n-D glasses” as a way to super-intelligence and virtual data scientists.
Future of Interpretability

- Creating simplified explainable models with prediction that humans can actually understand.
- “Downgrading” complex models for humans to understand by adding the explanation layer to the CNN and other deep learning models.
- Expanding visual and hybrid explanation models
- Developing explainable Graph Models
- Developing ML model in First Order Logic (FOL) terms of the domain ontology from which the data are used to build the model.
- Enhancing past probabilistic First Order Logic models to current much higher computational power. NP hard problems.
  - Previous models [Muggleton, 1991; Lavrac, Dzeroski, 1994; Kovalerchuk, Vityaev [1970s, 2000]
REFERENCES


Kovalerchuk B., Grishin V. Adjustable General Line Coordinates for Visual Knowledge Discovery in n-D data, Information Visualization, 18(1), 2019, pp. 3-32.

Kovalerchuk B., Grishin V. Adjustable General Line Coordinates for Visual Knowledge Discovery in n-D data, Information Visualization, 18(1), 2019, pp. 3-32.

Questions?

BORIS KOVALERCHUK

www.cwu.edu/~borisk

More information

borisk@cwu.edu