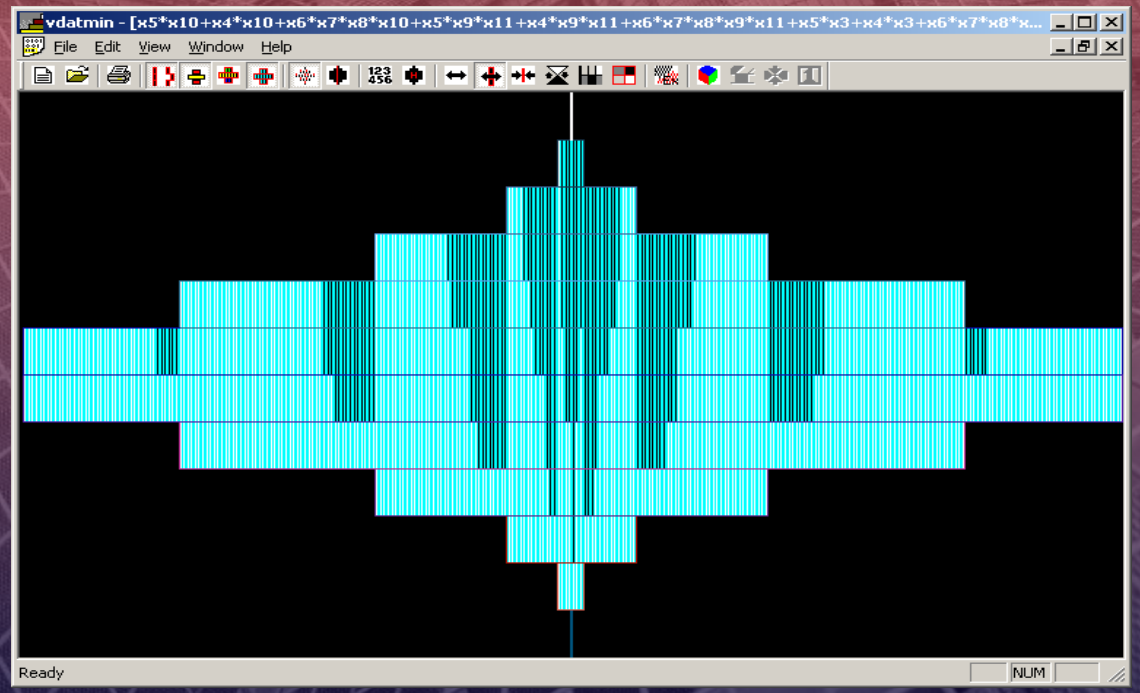
OPEN DATA SCIENCE CONFERENCE

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

The noblest pleasure is the joy of understanding.
Leonardo da Vinci

- 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

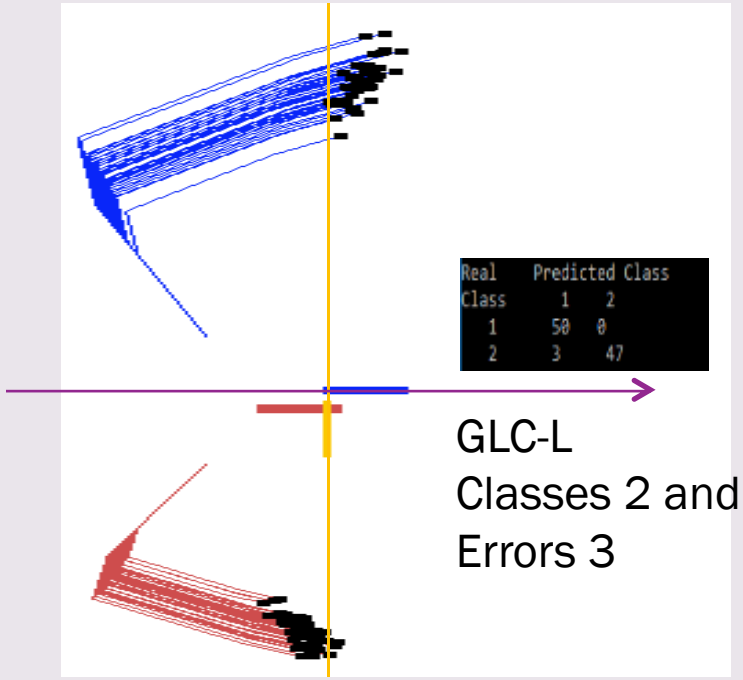
Setosa

Versicolor

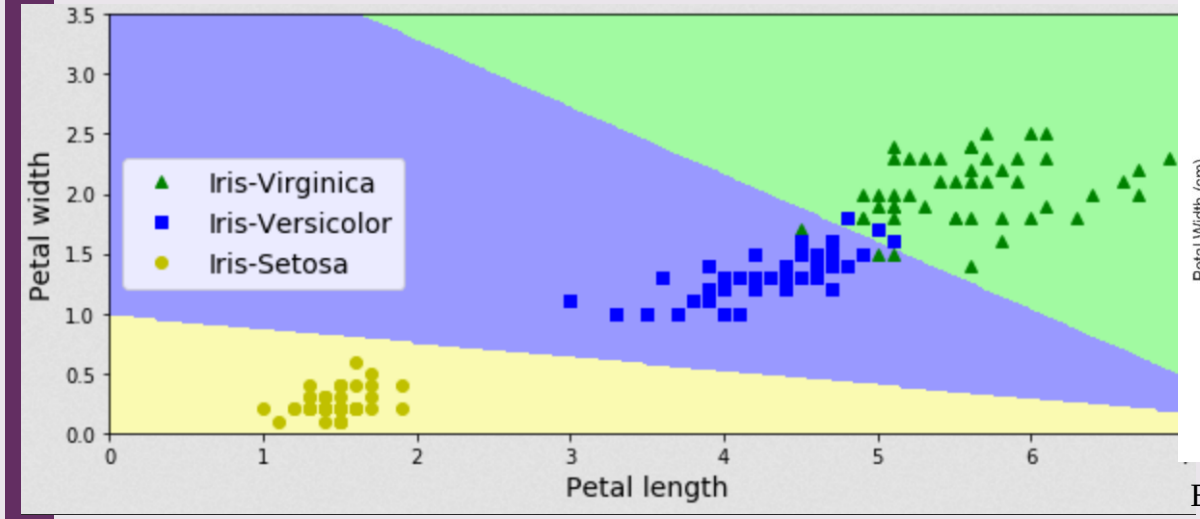
Virginica

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

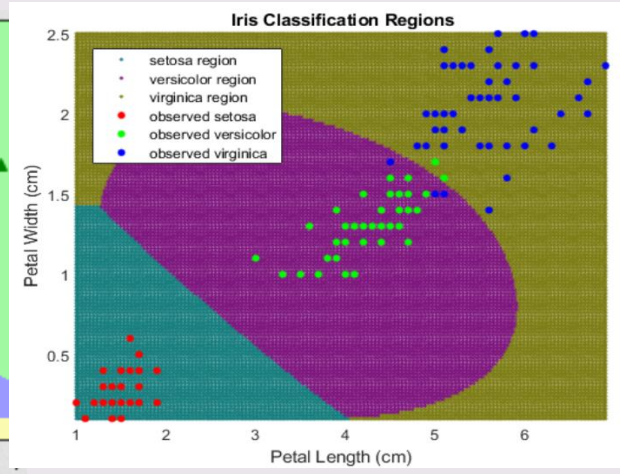


GLC-L
Classes 2 and 3
Errors 3



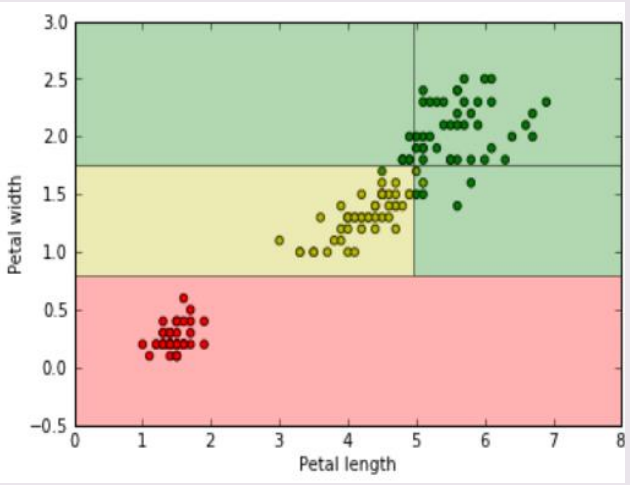
Example of logistic regression classification of these classes of Iris [Big data, 2018].

5 Errors



Example of SVM [Matlab, FITCSVM, 2018] and Decision Tree classification [Taylor, 2011] of th classes of Iris.

5 Errors



3 Errors

Human Visual and Verbal Generalization vs. Machine Learning generalization for Iris data

- 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].

- Kovalerchuk, 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.*

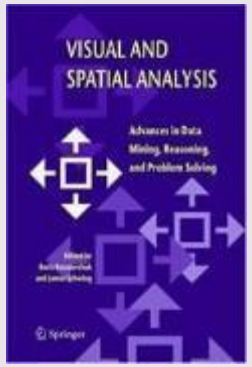
ML → VIS/EXP (1) (3) (4) ML first

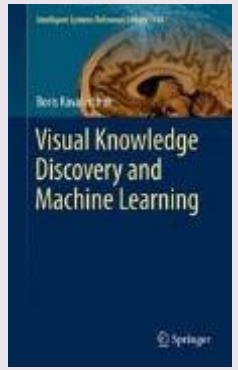
VIS → ML/EXP (2) (3) (5) Vis first

VIS-ML Joint Vis leads

ML-VIS Joint ML leads

- **Sources**: Multiple relevant publications including presenter's books:
  - *"Visual and Spatial Analysis: Advances in Visual Data Mining, Reasoning, and Problem Solving" (Springer, 2005), and*
  - *"Visual Knowledge Discovery and Machine Learning" (Springer, 2018).*





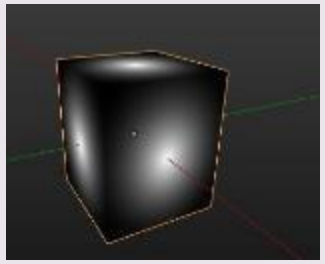
Dr. Boris Kovalerchuk

- 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

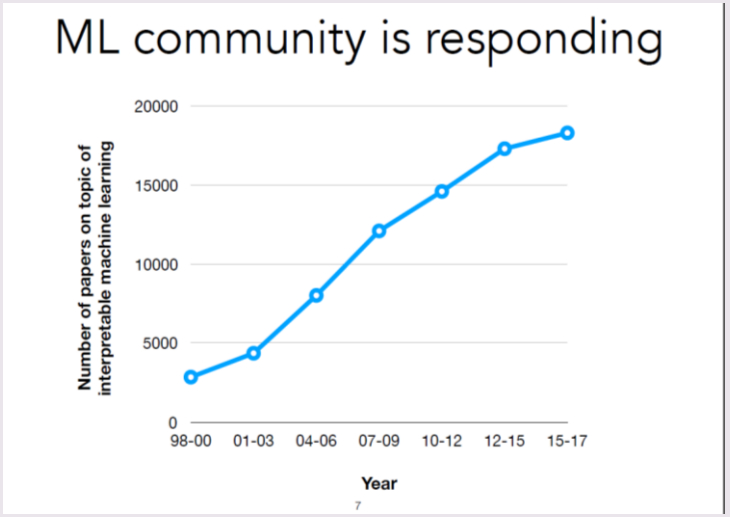
- Deep learning Neural networks
- Boosted trees
- Random forests
- ….

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**

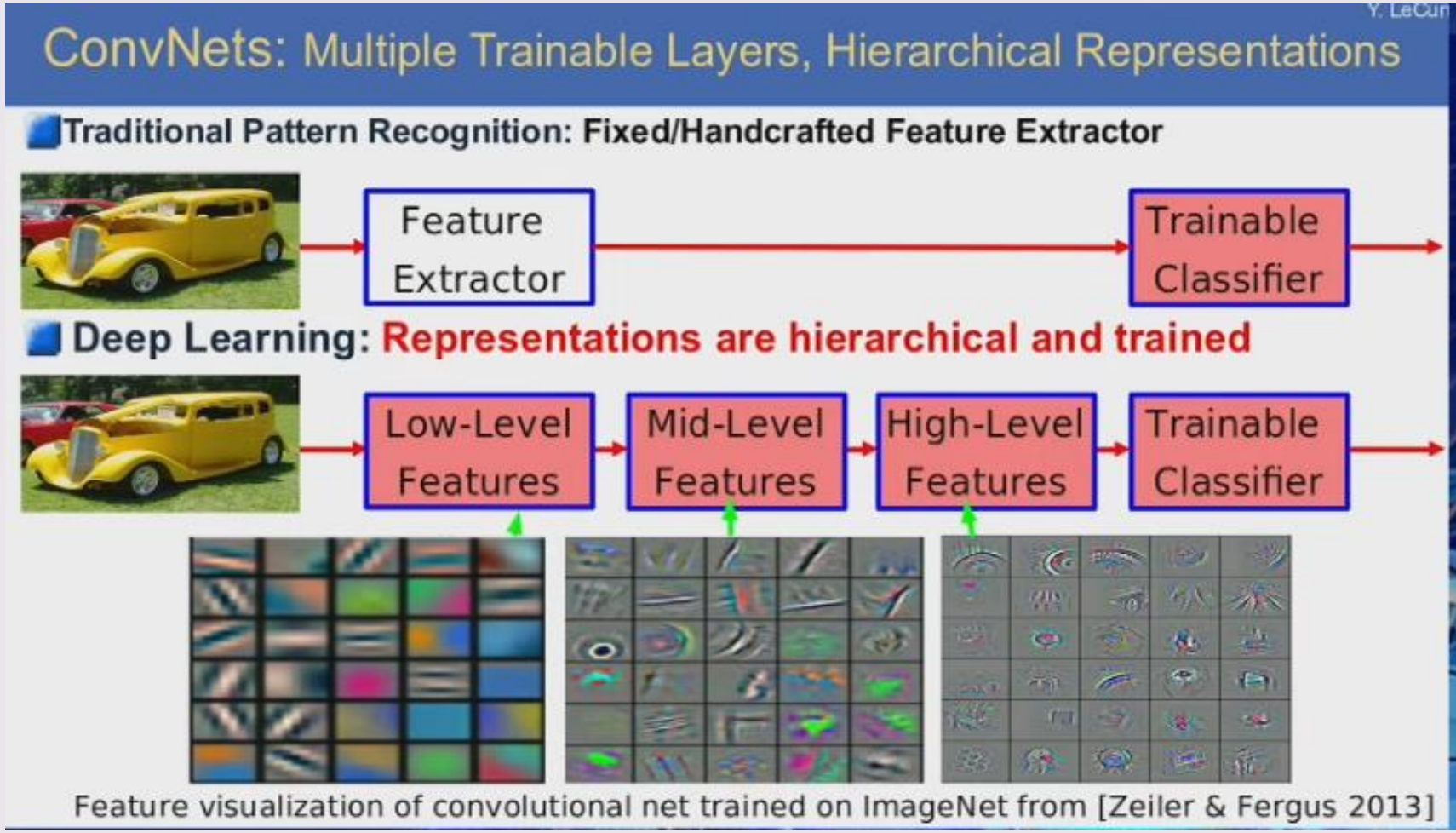
B.Kim, Introduction to Interpretable Machine Learning Tutorial, CVPR 2018, http://deeplearning.csail.mit.edu/slide_cvpr2018/been_cvpr18tutorial.pdf

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.*

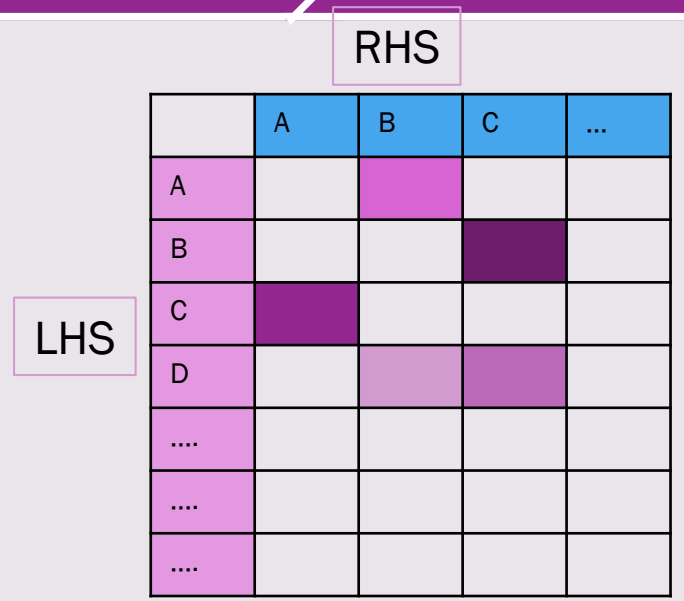
Demonstration of DNN models to understand them via heat maps visualizing discovered features



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.*

If A then B, A ⇒ B
If B then C
If C then A
If D then B
If D then C

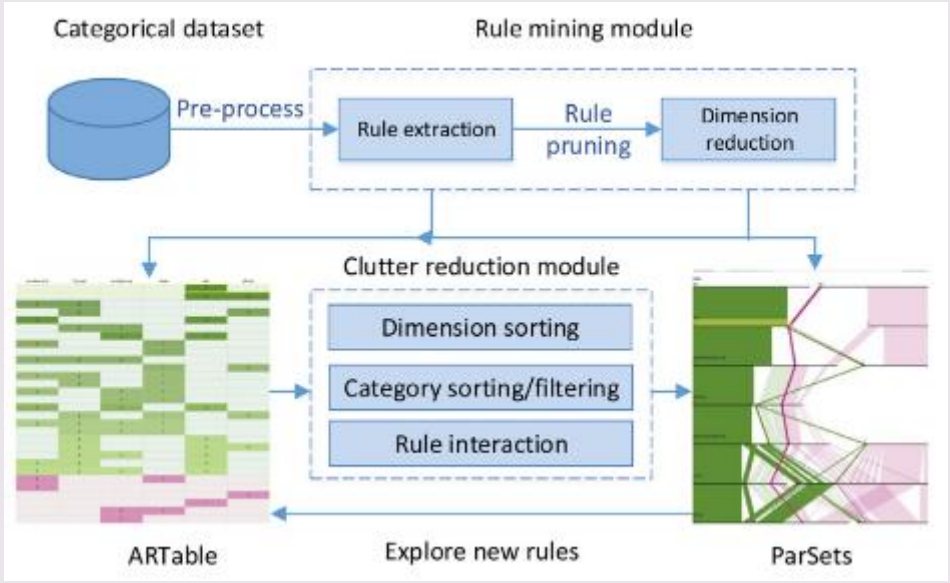


- Visualization approaches:
  - *Matrix with rows as Left Hand Side, LHS itemsets of rules and columns as Right Hand side (RHS) itemsets of rules.*
  - *Parallel sets*
- Challenges:
  - *Scalability for many LHS and RHS*
  - *readability of small cells having many categories in variables.*

Zhang C. et al. Association rule based approach to reducing visual clutter in parallel sets, Visual Informatics 3 (2019) 48–57

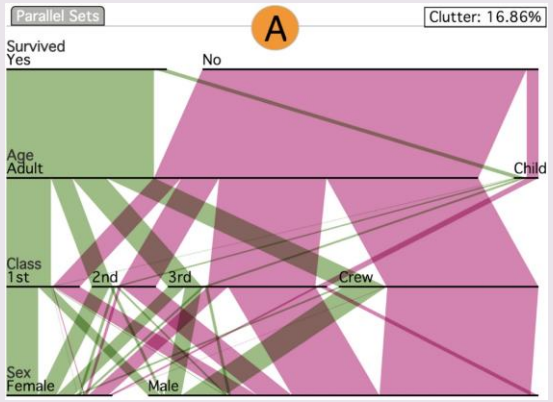
Visualization of Association Rules (AR) using Parallel sets for categorical data

- 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*

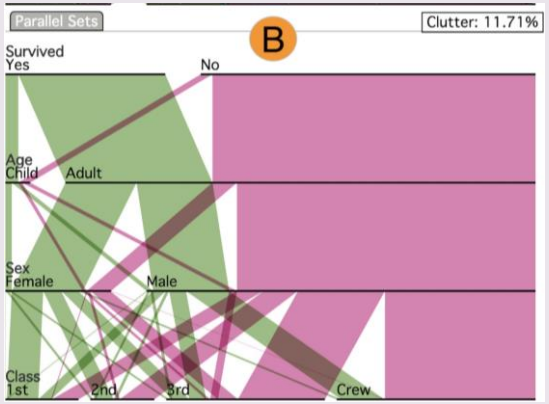


Zhang C. et al. Association rule based approach to reducing visual clutter in parallel sets, Visual Informatics 3 (2019) 48–5

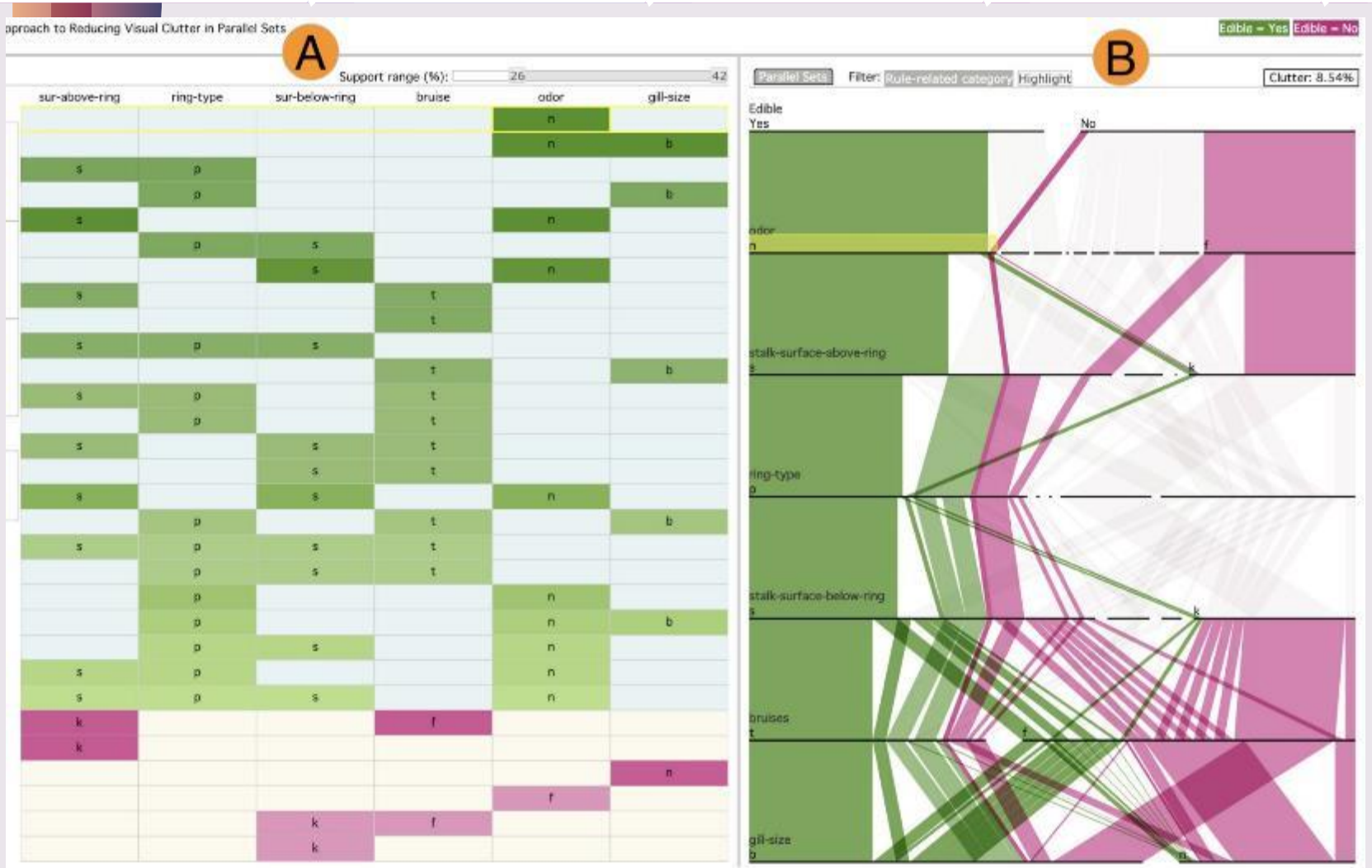
Before



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.



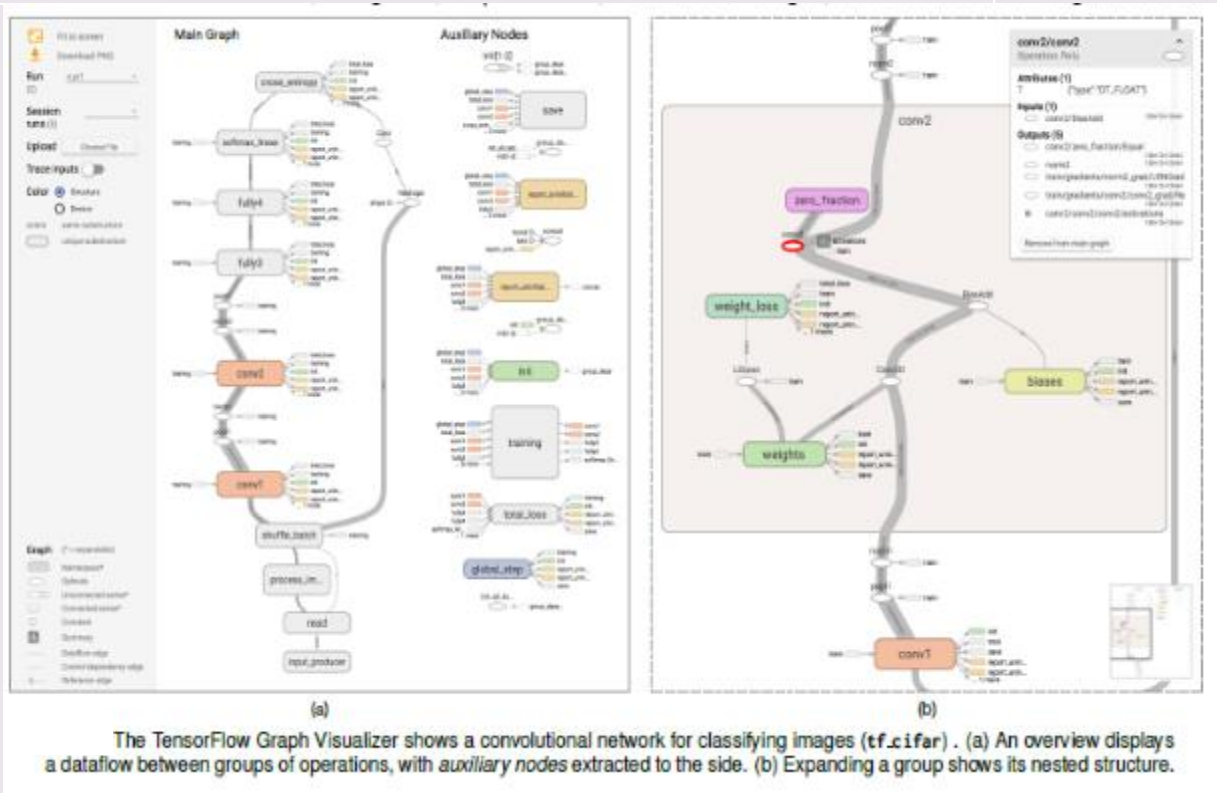
A: Association Rule Table (ARTable) -- the rules with sorting panel.
Columns – dimensions extracted from association rule result.
Rows -- rules with cells indicating the itemsets in the rule.
Color intensity expresses % or rule support..

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

Video 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)*



The TensorFlow Graph Visualizer shows a convolutional network for classifying images (`tf.cifar`). (a) An overview displays a dataflow between groups of operations, with *auxiliary nodes* extracted to the side. (b) Expanding a group shows its nested structure.

Wongsuphasawat K, Smilkov D, el al., Visualizing dataflow graphs of deep learning models in tensorflow. IEEE transactions on visualization and computer graphics. 2018: 24(1):1-2.

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*

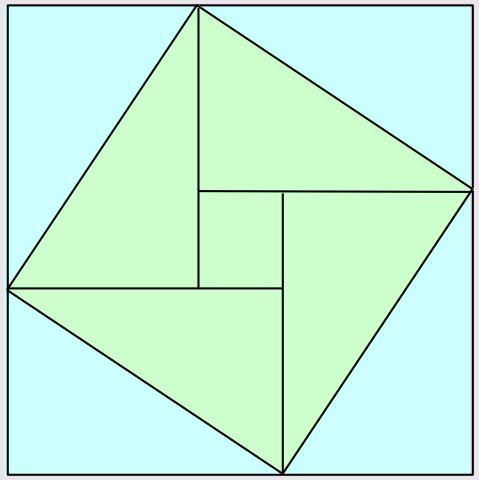
- Why Visual?
  - To leverage human perceptual capabilities
- Why interactive?
  - To leverage human abilities to adjust tasks on the fly
- Why Machine Learning?
  - To leverage analytical discovery that are outside of human abilities.
  - We cannot see patterns in multidimensional data by a naked eye.

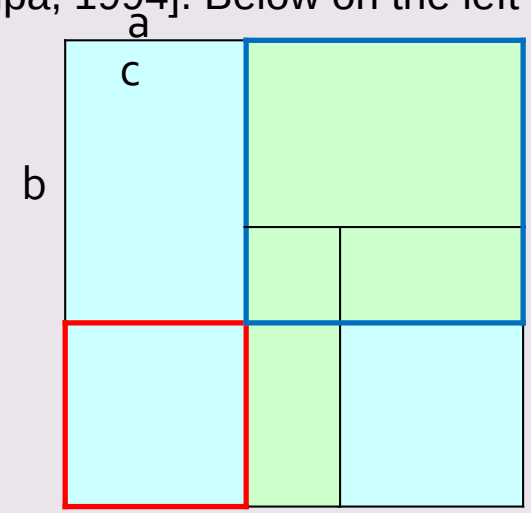
Visuals for creative thinking

- 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





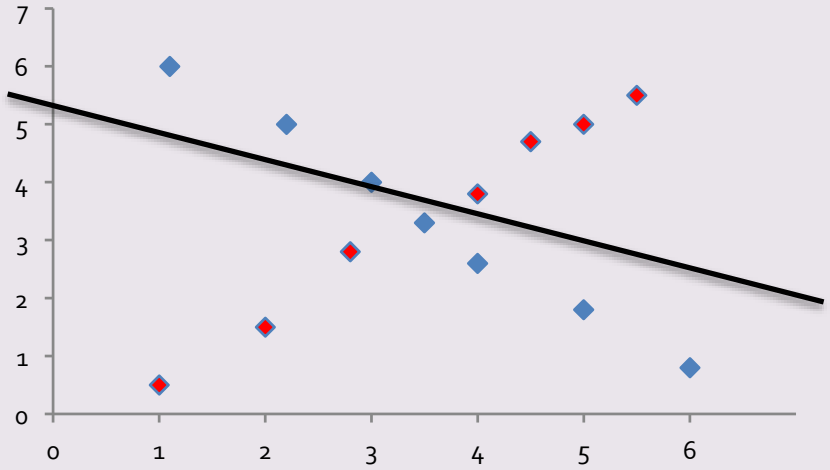
See

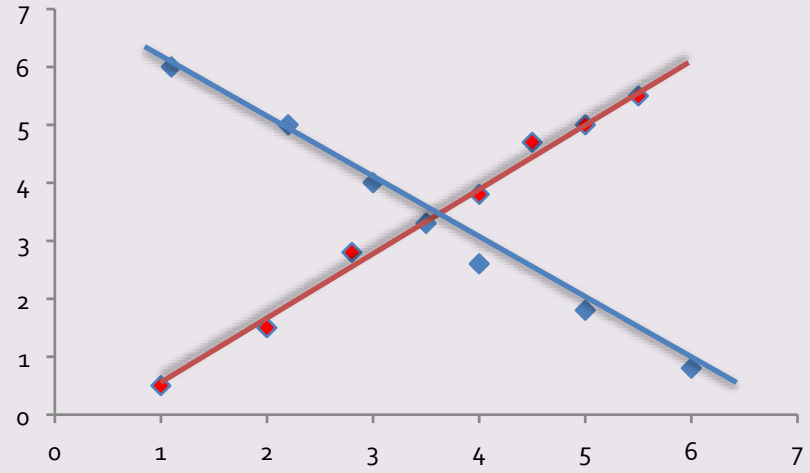
$(a+b)^2$ (area of the large square)= $a^2+b^2+ab+ab=(a+b)^2$

$a^2+b^2=(a+b)^2$(area of the large square) - 2ab (4 light green triangles) =c^2 (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





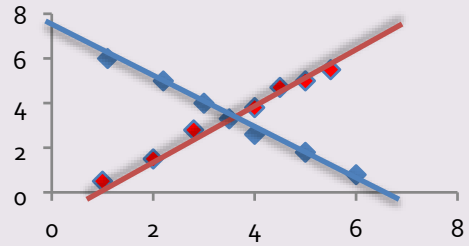
x	y	class
1	0.5	1
1.1	6	2
2	1.5	1
2.2	5	2
2.8	2.8	1
3	4	2
3.5	3.3	1
4	3.8	1
4	2.6	2
4.5	4.7	1
5	1.8	2
5	5	1
5.5	5.5	1
6	0.8	2

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.

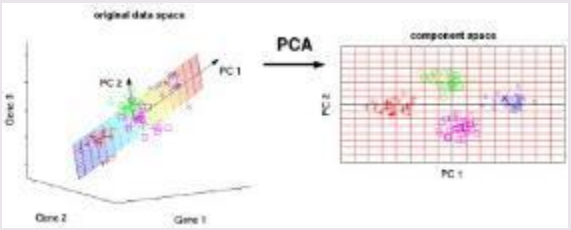
How to reproduce the success in 2-D for n-D data?



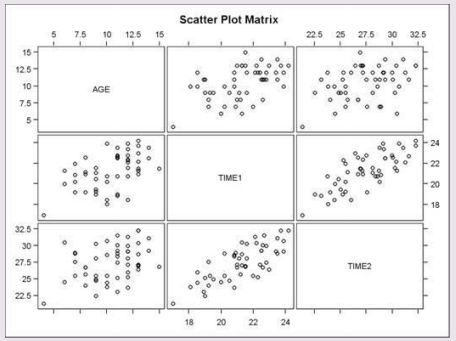
In high-dimensions we cannot see data with a **naked eye**.
We need methods for *lossless and interpretable visualization* of n-D data in 2-D.

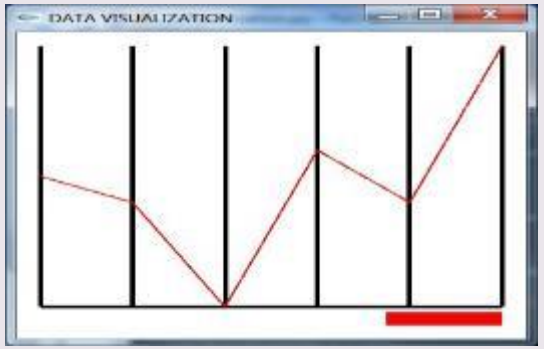
ID	FD1	FD2	FD4	FD5	FD6	FD10	FD12	FD15	FD16	FD18	FD20	FD22	FD23	FD24	FD25	FD26	FD27	FD28
1	0	0	2.749807	9.826302	4.067554	0	0	0	5.244006	0	2.743422	0	0	0	0	6.254963	0	0
2	11.51334	9.092989	0	12.46223	0	7.597155	0	0	8.940897	0	0	0	4.268456	0	0	0	0	1.309903
3	10.27931	0	2.075787	0	4.042145	0	0	0.477713	3.97378	0	0	2.477745	0	0	0	5.583099	0	7.418219
4	0	18.31495	0	0	0	0	0	0	4.472742	4.671682	0	7.248355	12.11645	0	0	0	6.030322	0
5	14.12261	15.1236	9.695051	0	0.915031	0	0	6.086389	9.139287	0	0	0	8.931774	0	0	0	0	0
6	0	0	5.405394	0	0	2.951092	0	3.797284	4.576391	0	0	0	0	0	2.763756	0	0	2.562996
7	0	0	0	8.068472	0	3.267916	0	0	5.09157	6.082168	0	0	5.42044	0	0	4.431955	0.415844	2.73227
8	6.169271	4.918356	5.566813	0	0	4.884737	5.168666	0	5.189289	0	0	0	2.49011	0	4.750784	2.994664	0	0
9	11.64548	0	0	12.16663	0	8.407408	0	0	0	0	0	0	4.289772	0	0	4.652006	0	0
10	9.957874	7.829115	0	0	0	0	0	0	7.082694	8.388349	0	0	0	0	0	4.706276	0	0.705345
11	9.994487	12.3192	3.058695	0	0	0	6.111047	0.380701	3.904454	0	2.573056	0	0	0	0	5.610187	0	0
12	0	8.446147	7.506574	0	0	5.846259	7.362241	6.557457	7.627757	9.05184	0	0	0	0	6.646436	0	0	0
13	13.65315	18.11681	2.457055	0	8.218276	0	5.689919	0	4.45029	3.213032	5.992753	0	11.56691	0	0	7.734966	0	0
14	0	0	0	8.710629	0	0	0	0	6.466624	0	0	0	3.865449	0	5.339944	3.943355	0	0
15	11.08665	0	0	12.57808	0	8.377558	0	9.269582	0	10.28637	0	0	4.141793	0	0	4.953615	0	0.433766
16	0	0	7.32989	9.848915	0	0	6.639803	0	0	0	0	0	0	0	0	4.288343	0	0
17	0	0	8.49376	0	0	0	7.403671	9.346368	0	0	0	0	0	0	0	0	0	0
18	9.52255	0	0	10.30969	0	0	6.508697	0	0	9.04743	0	0	3.113288	0	7.667032	0	0	0
19	0	9.237608	3.488988	7.443493	0	0	0	0	0	0	0.921821	1.305681	0	0	0	4.447716	0	4.174564
20	0	16.78071	2.745921	0	5.606468	0	7.824948	0	0	4.807075	4.454489	0	0	0	0	7.226364	0	10.62363
21	0	0	8.18506	0	0.469365	4.241147	0	5.823779	0	0	0	0	0	0	0	6.475445	0	4.49432
22	9.609696	12.07202	0	6.483721	0	0	0	0	0	1.554688	0	5.446015	0	0	0	0	0	9.85667
23	10.71318	0	0	11.44685	0	8.097867	0	8.832153	8.646919	0	0	0	0	0	0	4.705225	0	0
24	6.625456	0	3.686915	6.715843	0.187058	0	3.735899	3.55698	0	0	0	0	0	0	2.996381	3.700704	0	0
25	9.794333	0	0	9.788224	0	4.599581	0	0	0	0	0	0	0	0	0	4.694789	0	910
26	10.25995	0	0	9.531824	0	1.156152	6.604298	0	0	0	0	0	6.346496	0	1.300262	0	1.869395	4.265034

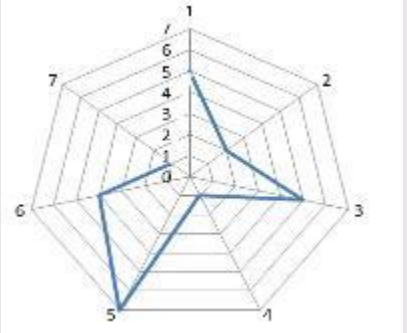
Multidimensional Visualization Approaches



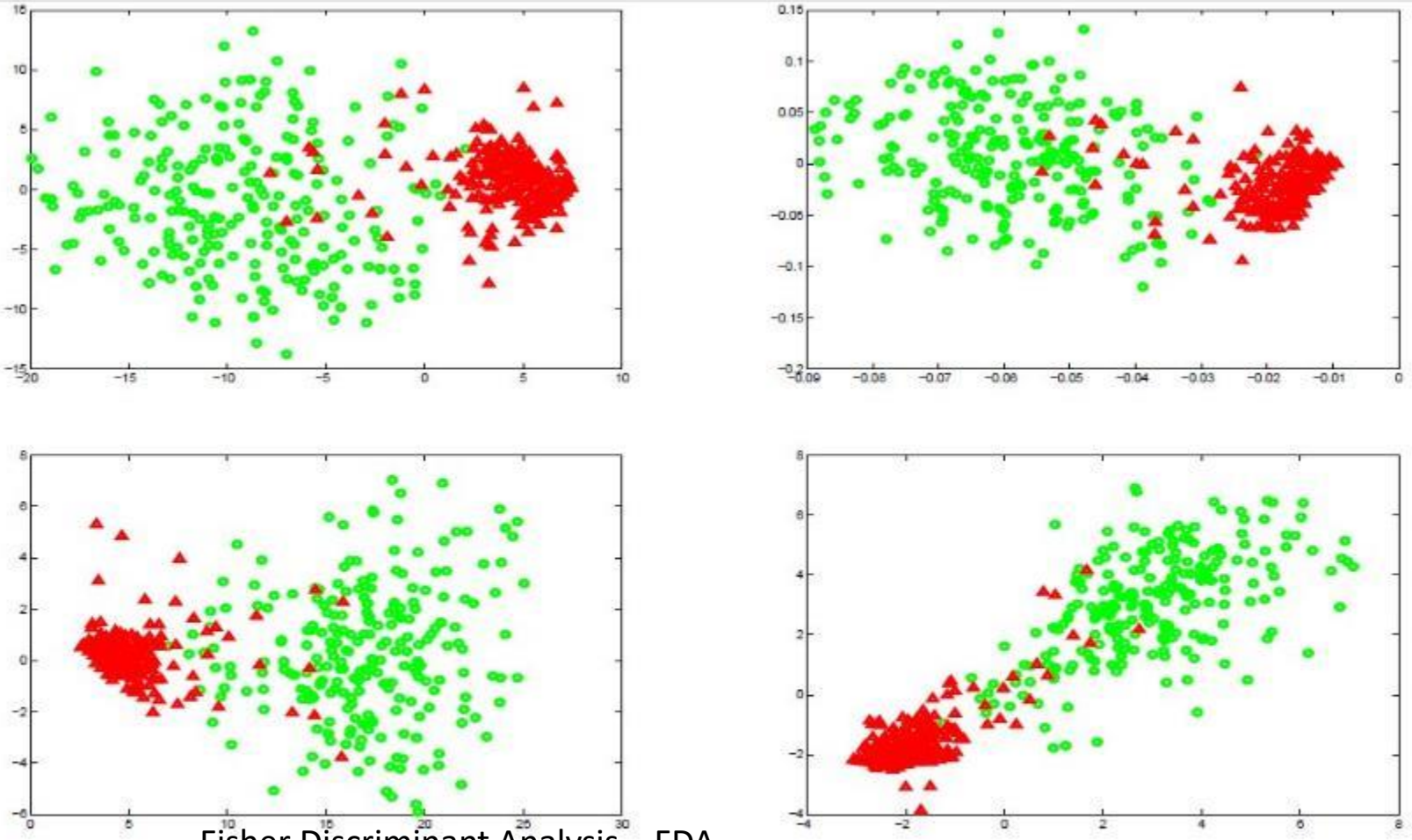
- The goal of multivariate, multidimensional visualization is representing *n-tuples (n-D points*) in 2-D or 3-D, e.g., (3,1,0.5, 7, 1.19, 0.13, 8.1)
- Often multidimensional data are visualized by lossy dimension reduction (e.g., PCA) or by splitting n-D data to a set of low dimensional data (pairwise correlation plots).
  - While splitting is useful it destroys integrity of n-D data, and leads to a shallow understanding complex n-D data.
- To mitigate splitting difficulty
  - an additional and difficult perceptual task of assembling 2-dimensional visualized pieces to the whole n-D record must be solved.
- An alternative way for deeper understanding of n-D data is
  - developing visual representations of n-D data in low dimensions without such data splitting and loss of information **as graphs not 2-D points.**
  - E.g., Parallel and Radial coordinates.
  - Challenge -- occlusion







WBC data



Fisher Discriminant Analysis – FDA

g. 4. Wisconsin data set, top row: MDS and PCA, bottom row: FDA and SVM

Benign and malignant cancer cases overlap

Interpretation of dimensions is difficult. Non-reversible lossy methods: 9-D to 2-D.

T. Maszczyk and W. Duch Support Vector Machines for visualization and dimensionality reduction, LNCS, Vol. 5163, 346-356, 2008

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 R^n, and a number $k > 8\ln(m)/\varepsilon^2$, there is a linear map $f : R^n \longrightarrow R^k$ 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 ε.**

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 R^n there is a mapping f: $R^n \rightarrow R^k$ with "similar" distances in R^n and R^k between mapped points. This similarity is expressed in terms of error 0 < ε < 1.*
- *For ε = 0 these distances are equal. For ε=1 the distances in R^k are less or equal to $\sqrt{2}S$, where S is the distance in R^n. This means that distance s in 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< ε <1 and any integer n, let k be a positive integer such that*

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

- *then for any set V of m points in R^k there is a mapping f: $R^n \rightarrow R^k$ such that for all u, v $\in V$*

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

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

$$k = \lceil 9(\varepsilon^2 - 2\varepsilon^3/3)^{-1} \ln n \rceil + 1 \qquad (3)$$

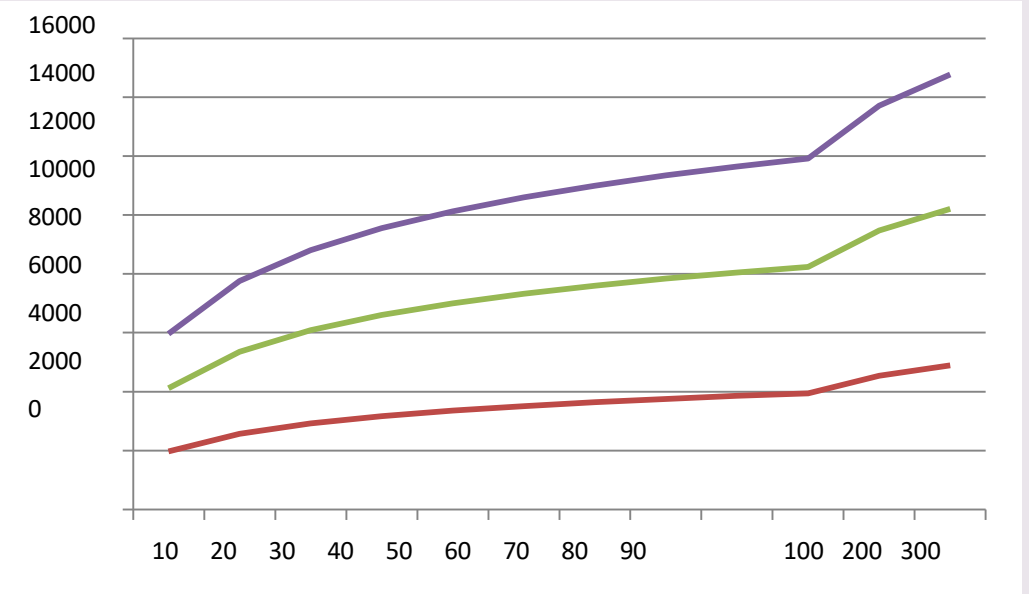
Theoretical limits to preserve n-D distances in 2-D: Johnson-Lindenstrauss Lemma

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.

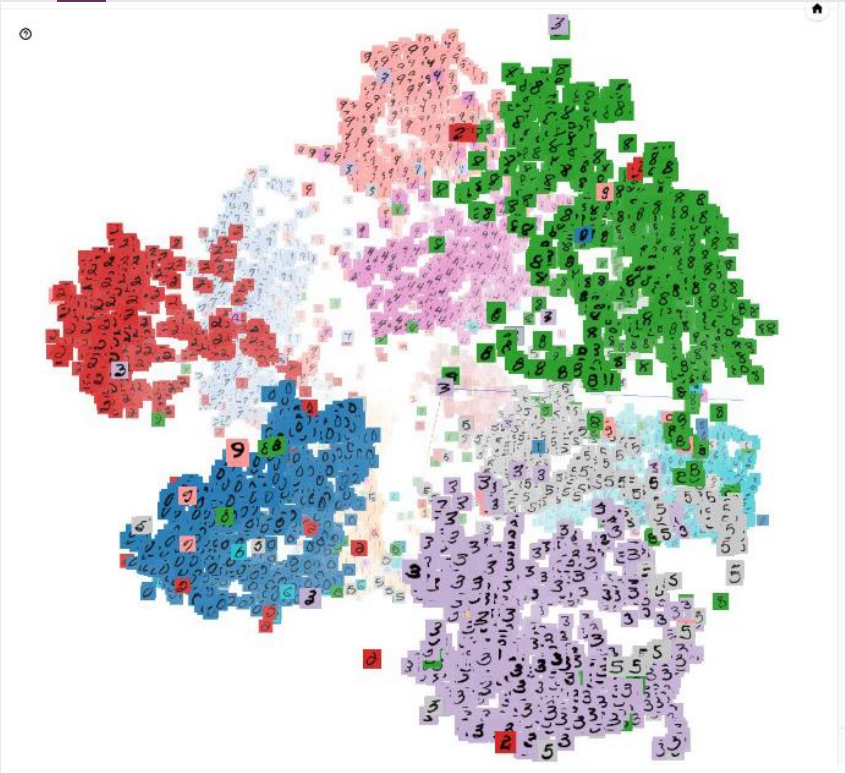
Number of arbitrary points in high-dimensional space	Sufficient dimension with formula (1)	Sufficient dimension with formula (2)_	Insufficient dimension with formula (3)
10	1974	2145	1842
20	2568	2791	2397
30	2915	3168	2721
40	3162	3436	2951
50	3353	3644	3130
60	3509	3813	3275
70	3642	3957	3399
80	3756	4081	3506
90	3857	4191	3600
100	3947	4289	3684
200	4541	4934	4239
300	4889	5312	4563

Dimensions to support ±31% of error (ε=0.1).



Dimensions required supporting ±31% of error ε.

TensorFlow Embedding projector and warnings



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)".

Choo J, Liu S. Visual analytics for explainable deep learning. IEEE computer graphics and applications. 2018 Jul 3;38(4):84-92.

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].

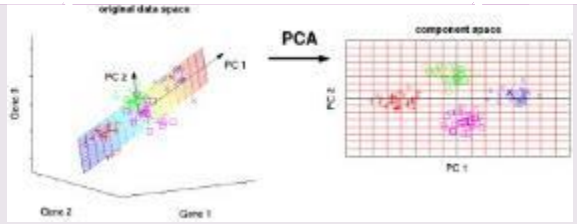
- PCA can ***distort local neighborhoods***.
- t-SNE can ***distort global structure*** trying to preserve local data neighborhoods
- t-SNE may not help to find ***outliers*** or assign meaning to ***densities*** in clusters.
- Outlier and dense areas *visible in t-SNE may not be them* in the n-D space.
- 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***.

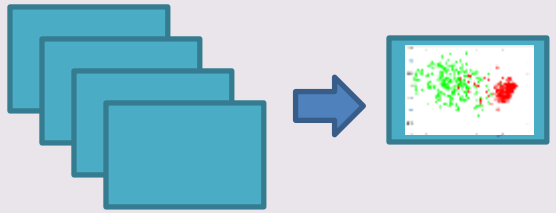
- Therefore, we focus on ***point-to-graph GLC methods***, which open a new opportunity to address this deficiency – *loss of n-D structure and properties*.
- However, we can see the statements that users **can easily identify outliers** in t-SNE. It will be valid only by showing that the n-D data similarity 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).

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.

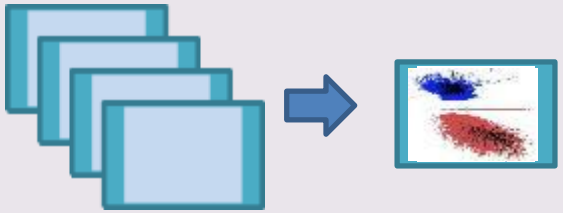
2. Discover n-D patterns in n-D data then visualize them in 2-D as graphs.
3. Join 1 and 2: some patterns are discovered in (1) with controlled errors and some are discovered in (2).





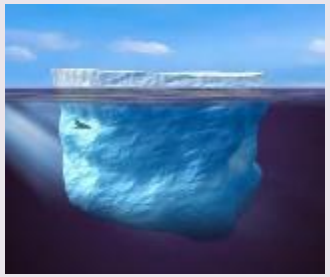
n-D data

2-D data & 2-D patterns

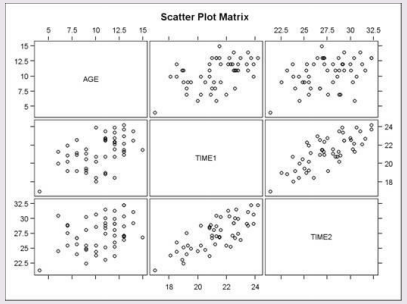


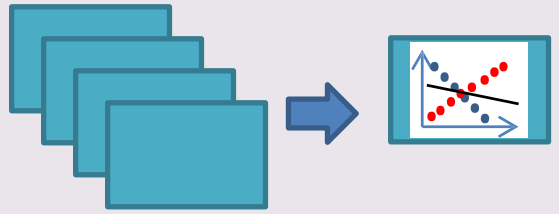
n-D data and n-D patterns

2-D data & n-D patterns



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

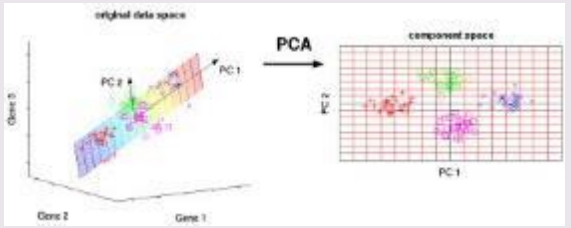


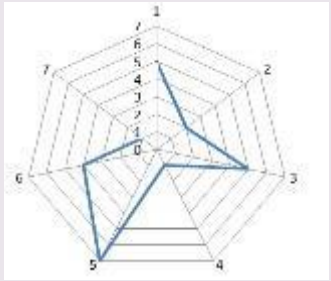


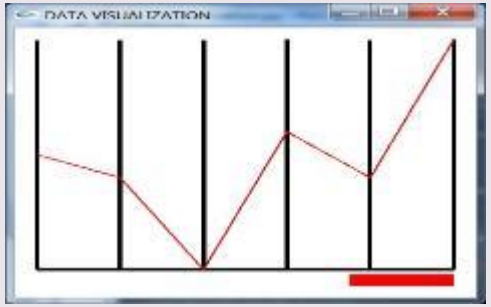
n-D data

2-D data with 2-D patterns

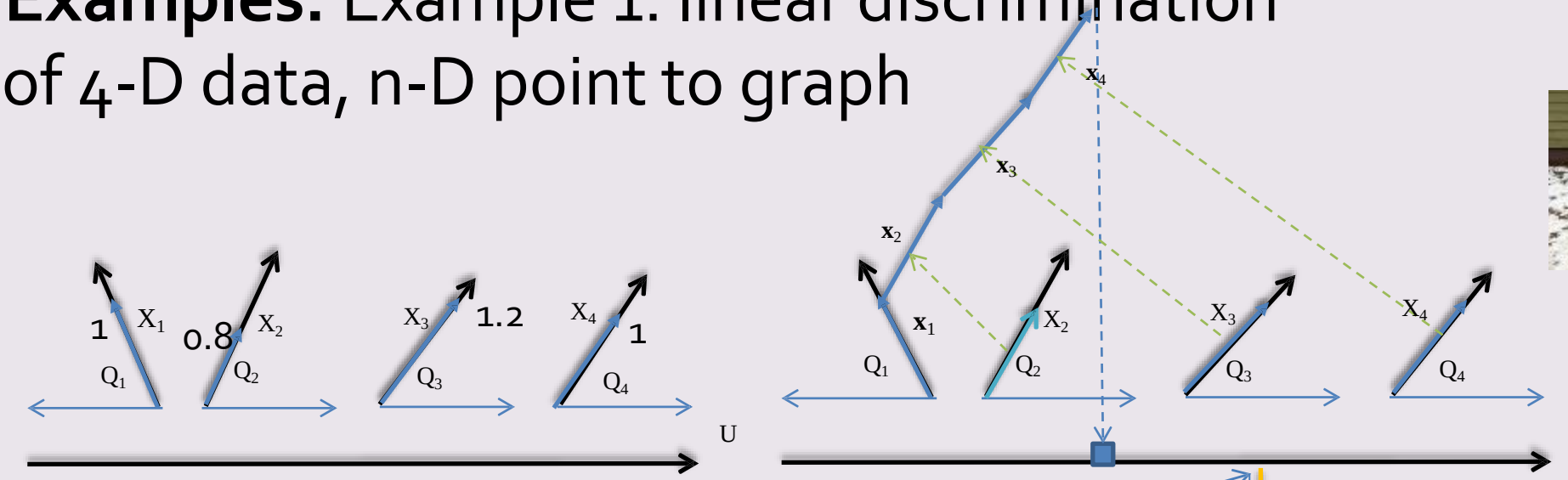
- Lossy approach
  - Lossy conversion to 2-D (dimension reduction, DR)
    -- Point to point **(n-D point to 2-D point)**
  - Visualization in 2-D
  - Interactive discovery of 2-D patterns in visualization
- Lossless approach
  - Lossless conversion (visualization) to 2-D (n- D data fully restorable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

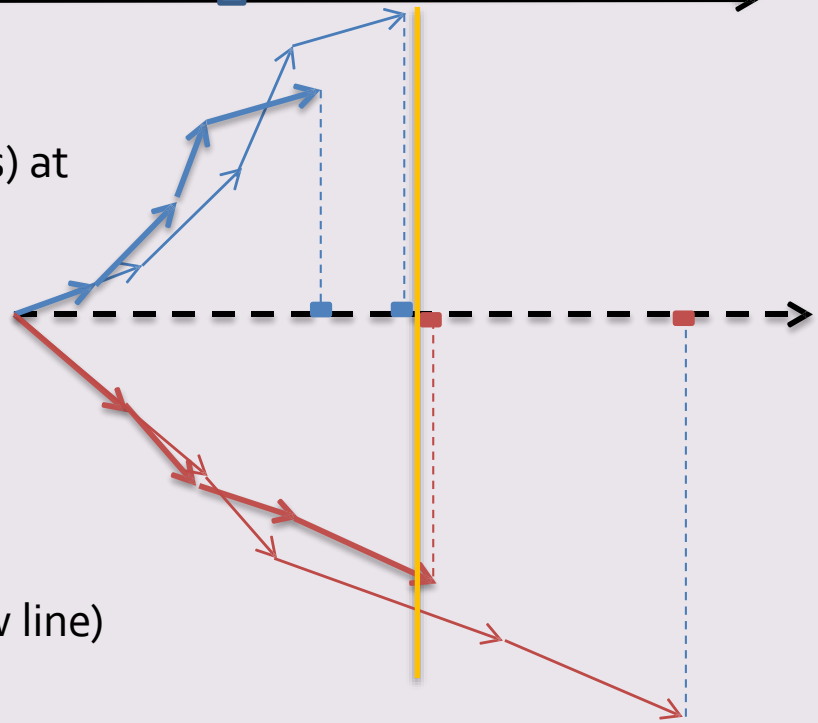


Algorithm GLC-L

- 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)

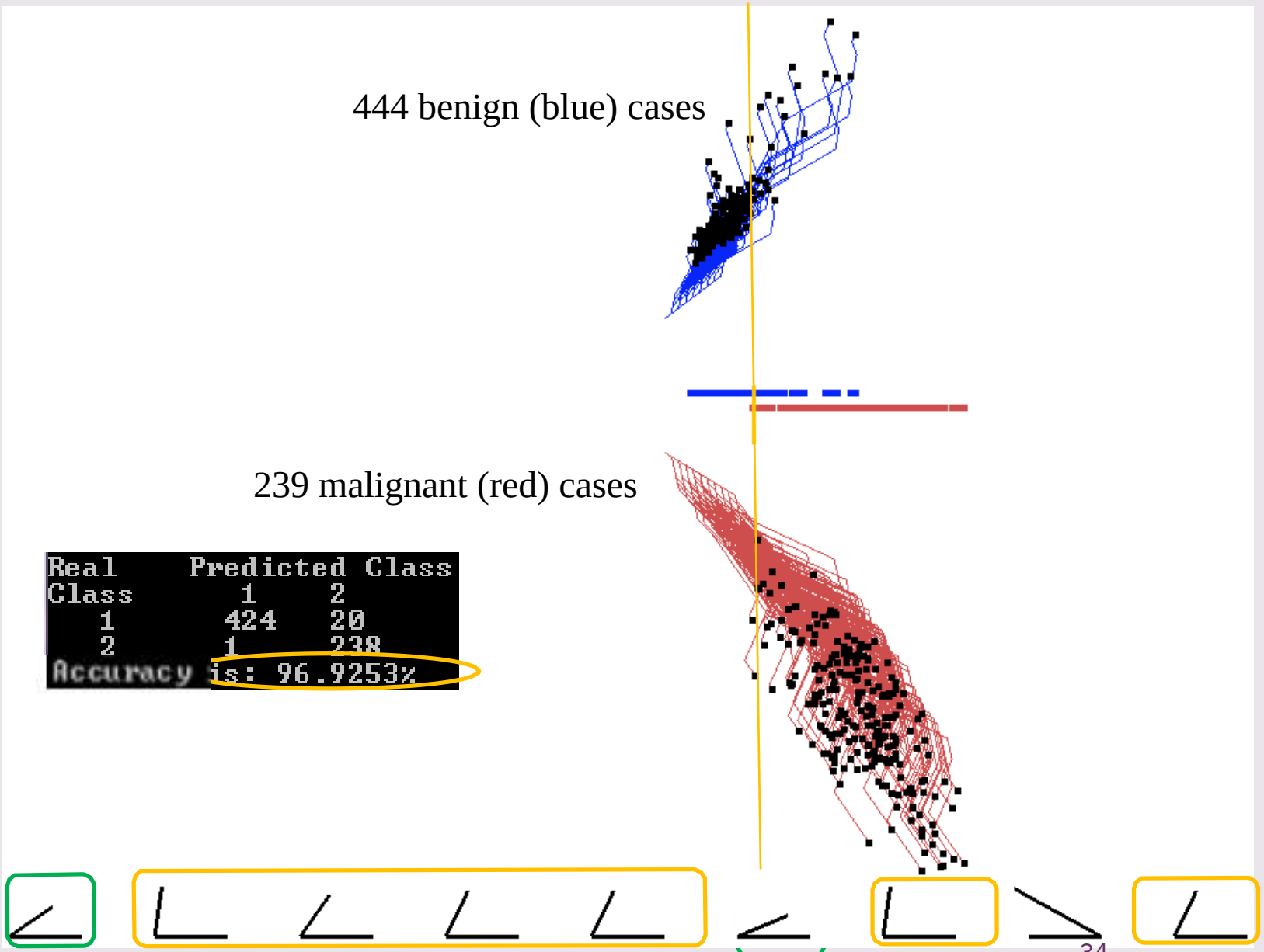


Example 2: Visual linear discrimination of 9-D Wisconsin Breast Cancer data in GLC-L coordinates

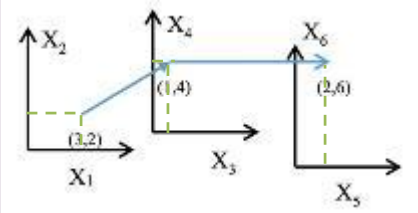
- Critical in Medical diagnostics and many other fields:
  - Explanation of patterns and
  - Understanding them
  - Lossless visual means
  - Reversible/restorable

Only one malignant (red case) on the wrong side

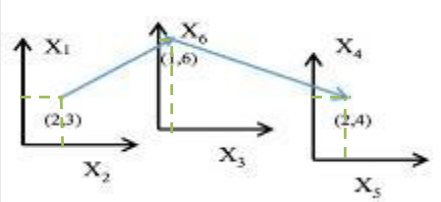
Angles Q_1-Q_9 – contributions of attributes X_1-X_9



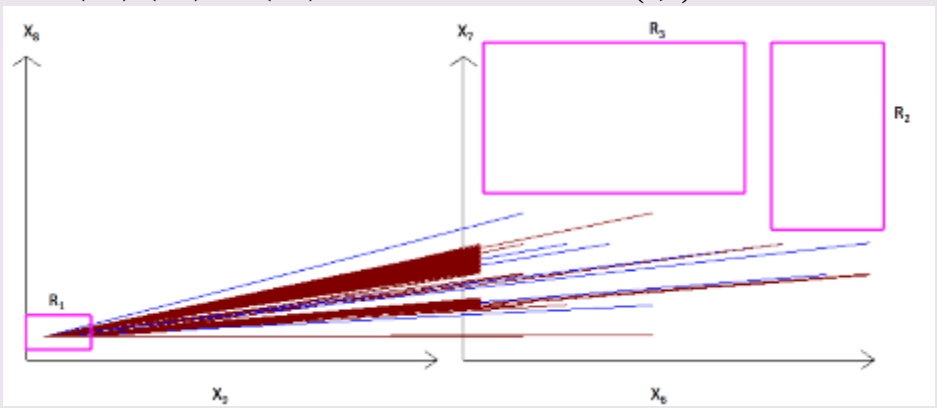
Example 3: Visual discrimination of 9-D Wisconsin Breast Cancer data in Shifted Paired Coordinates (SPC)



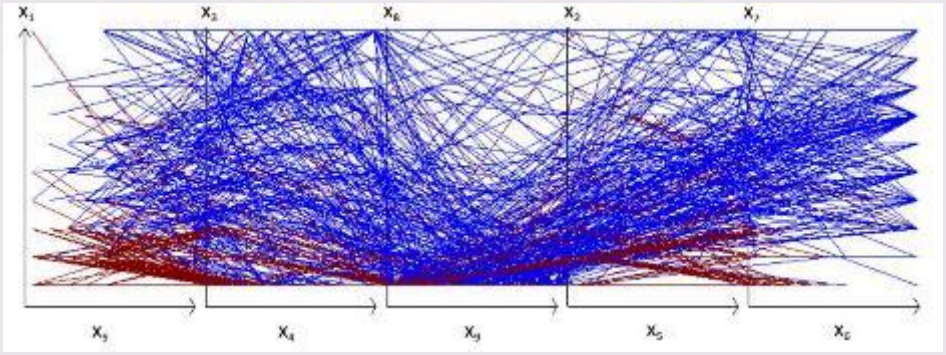
Point **a** in (X_1,X_2), (X_3,X_4), (X_5,X_6) as a sequence of pairs (3,2), (1,4) and (2,6).



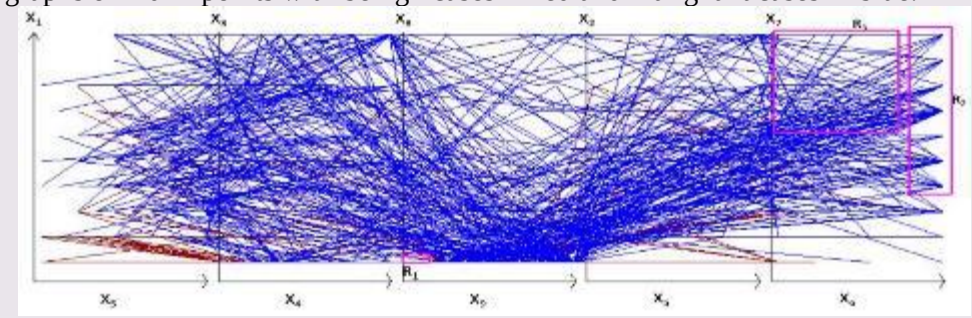
Point **a** in (X_2,X_1), (X_3,X_6), (X_5,X_4) as a sequence of pairs (2,3), (1,6) and (2,4).



WBC data in 4-D SPC as graphs in coordinates (X_9,X_8) and (X_6,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.



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

Here malignant cases (blue cases)

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*

Example 4: Avoiding Occlusion with Deep Learning on WBC data

Numeric 9-D Wisconsin Breast Cancer (WBC) Data

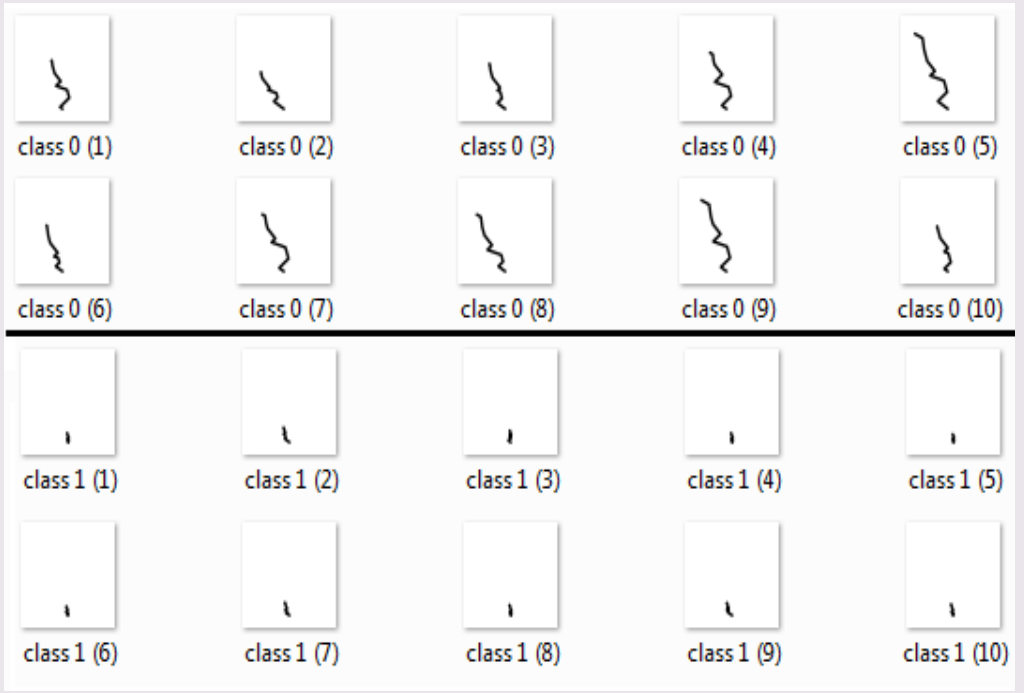
Visualized numeric data as downscaled 25x25 pixels images using GLC-L method

Deep Learning Convolutional Neural Network (CNN) on **images** (GLC-L **visualization**) as input

Classification accuracy 97.22% at the level and above published in literature

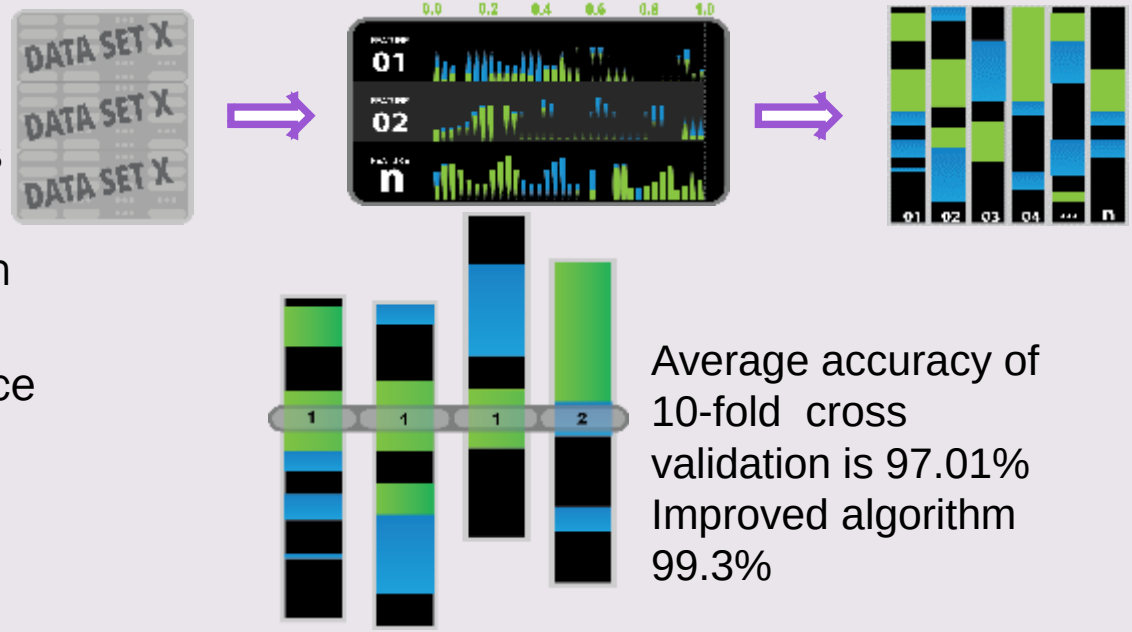
10-fold cross validation

Visualization optimization



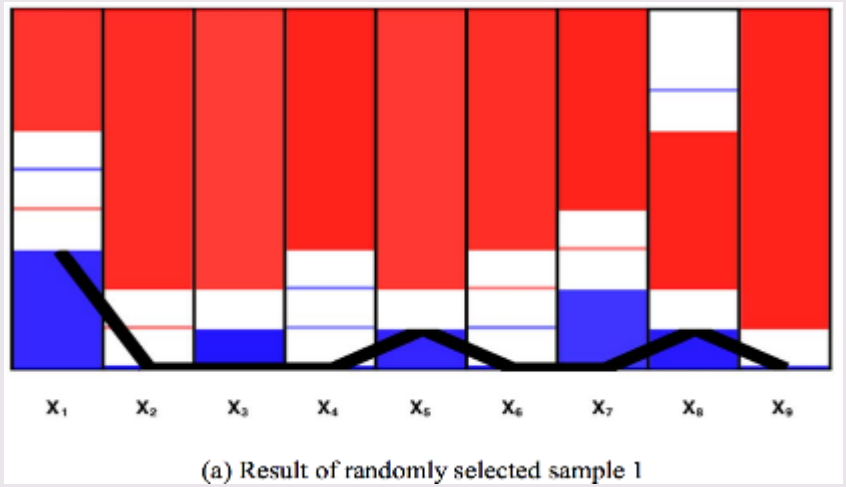
WBC data samples visualized in GLC-L for CNN model with the best accuracy.

EXAMPLE 5: INTERPRETABLE DCP ALGORITHM ON WBC DATA

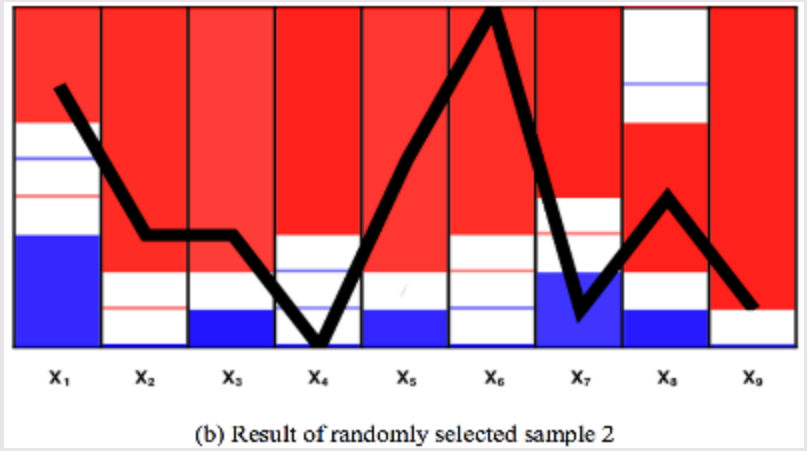


1. Constructing dominance intervals for classes on each attribute
2. Combining dominance intervals in the voting methods
3. Learning parameters of dominance intervals and voting methods for prediction
4. Visualizing the dominance structure
5. Explaining the prediction

Average accuracy of 10-fold cross validation is 97.01%
Improved algorithm 99.3%



(a) Result of randomly selected sample 1

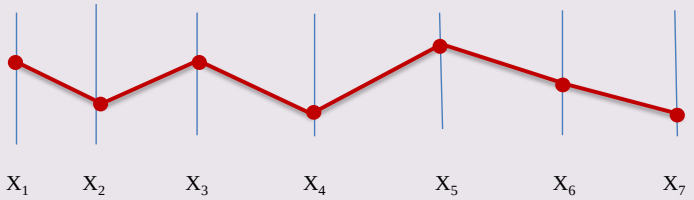


(b) Result of randomly selected sample 2

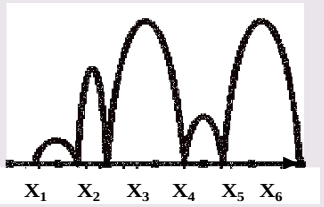
General Line Coordinates (GLC) to convert n-D points to graphs in 2-D

General Line Coordinates (GLC)

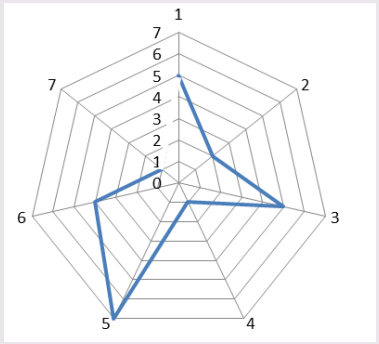
Descartes lay in bed and invented the method of co-ordinate geometry.
Alfred North Whitehead



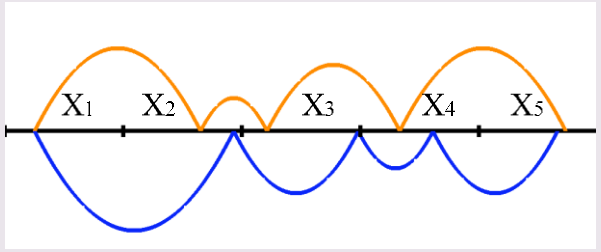
7-D point *D*=(5,2,5,1,7,4,1) in Parallel Coordinates



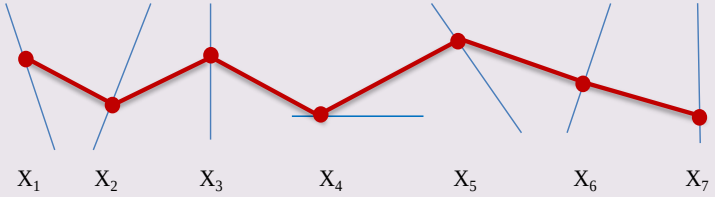
6-D (5,4,0,6,4,10) point in In-line Coordinates



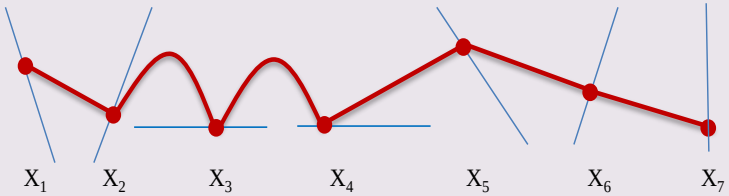
7-D point D=(5,2,5,1,7,4,1) in Radial Coordinates.



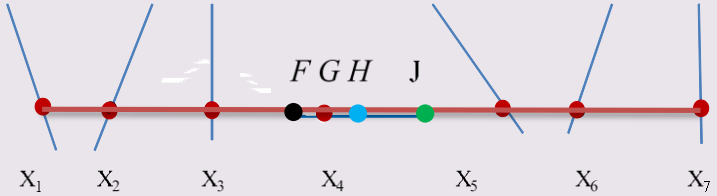
Two 5-D points of two classes in Sequential In-Line Coordinates.



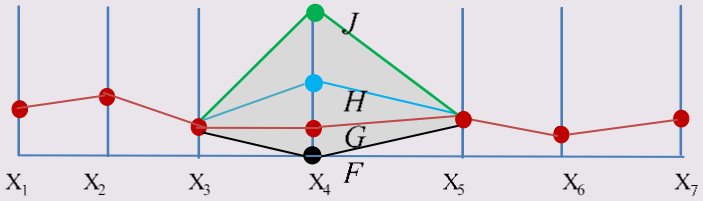
(a) 7-D point D in General Line Coordinates with straight lines.



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



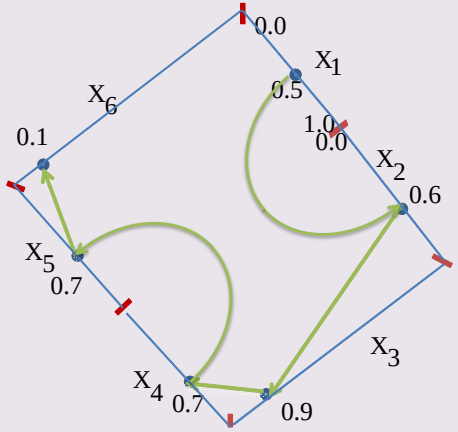
(c) 7-D points *F*-*J* in General Line Coordinates that form a simple single straight line.



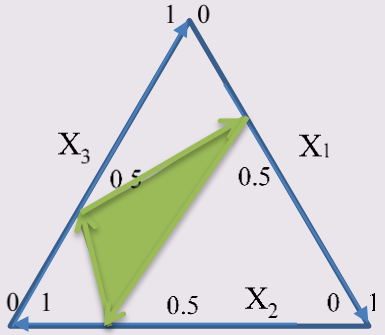
(d) 7-D points *F*-*J* in Parallel Coordinates that do not form a simple single straight line.

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

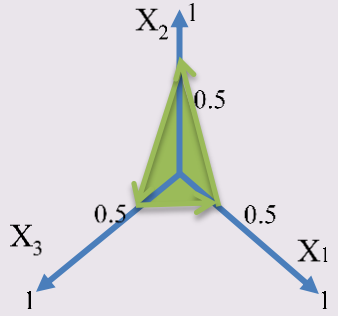
General Line Coordinates (GLC)



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

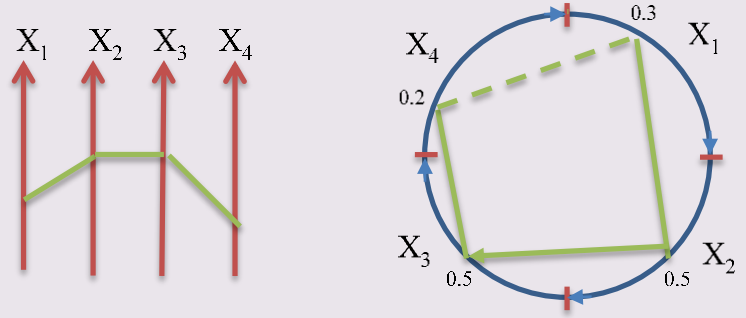


(a) Point A in in 3-Gon coordinates.



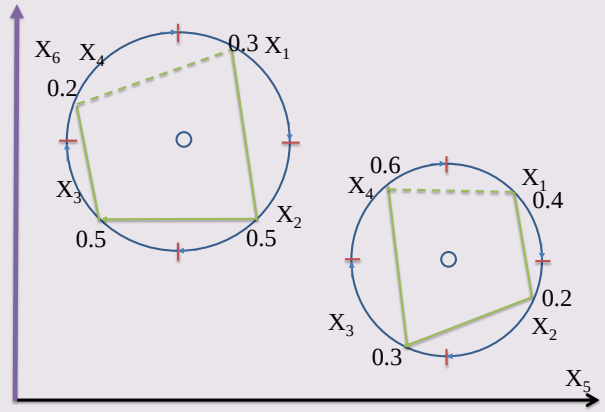
(b) Point A in in radial coordinates.

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



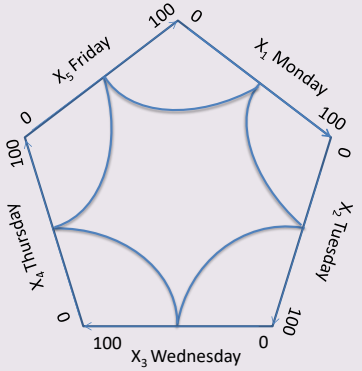
(a) Parallel Coordinates display.

(b) Circular Coordinates display.

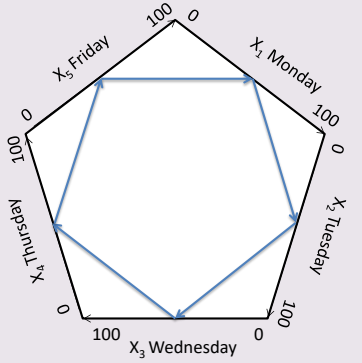


(c) Spatially distributed objects in circular coordinates with two coordinates X_5 and X_6 used as a location in 2-D and X_7 is encoded by the sizes of circles.

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



(a) Example in n-Gon coordinates with curvilinear edges of a graph.



(b) Example in n-Gon coordinates with straight edges of a graph.

Figure 2.6 Example of weekly stock data in n-Gon (pentagon) coordinates.

General Line Coordinates (GLC): 2-D visualization

2-D Line Coordinates.

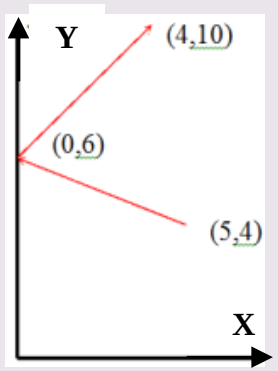
Type	*Characteristics*
2-D General Line Coordinates (GLC)	Drawing n coordinate axes in 2-D in variety of ways: curved, parallel, unparalleled, collocated, disconnected, etc.
Collocated Paired Coordinates (CPC)	Splitting an n-D point x into pairs of its coordinates $(x_1,x_2),\ldots,(x_{n-1},x_n)$; drawing 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.
Basic Shifted Paired Coordinates (SPC)	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.
2-D Anchored Paired Coordinates (APC)	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.
2-D Partially Collocated Coordinates (PCC)	Drawing some coordinate axes in 2D collocated and some coordinates not collocated.
In-Line Coordinates (ILC)	Drawing all coordinate axes in 2D located one after another on a single straight line.
Circular and n-Gon coordinates	Drawing all coordinate axes in 2D located on a circle or an n-Gon one after another.
Elliptic coordinates	Drawing all coordinate axes in 2D located on ellipses.
GLC for linear functions (GLC-L)	Drawing all coordinates in 2D dynamically depending on coefficients of the linear function and value of n attributes.
Paired Crown Coordinates (PWC)	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.

General Line Coordinates (GLC): 3-D visualization

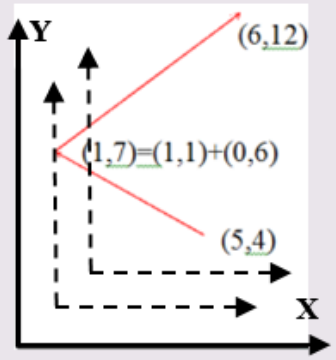
3-D Line Coordinates.

Type	*Characteristics*
3-D General Line Coordinates (GLC)	Drawing *n* coordinate axes in 3-D in variety of ways: curved, parallel, unparalleled, collocated, disconnected, etc.
Collocated Tripled Coordinates (CTC)	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.
Basic Shifted Tripled Coordinates (STC)	Drawing each next triple in the shifted coordinate system by adding (1,1,1) to the second tripple, (2,2,2) to the third tripple (*i*-1, *i*-1,*i*-1) to the *i*-th triple, and so on. More generally, shifts can be a function of some parameters.
Anchored Tripled Coordinates (ATC) in 3-D	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.
3-D Partially Collocated Coordinates (PCC)	Drawing some coordinate axes in 3-D collocated and some coordinates not collocated.
3-D In-Line Coordinates (ILC)	Drawing all coordinate axes in 3D located one after another on a single straight line.
In-Plane Coordinates (IPC)	Drawing all coordinate axes in 3D located on a single plane (2-D GLC embedded to 3-D).
Spherical and polyhedron coordinates	Drawing all coordinate axes in 3D located on a sphere or a polyhedron.
Ellipsoidal coordinates	Drawing all coordinate axes in 3D located on ellipsoids.
GLC for linear functions (GLC-L)	Drawing all coordinates in 3D dynamically depending on coefficients of the linear function and value of n attributes.
Paired Crown Coordinates (PWC)	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.

Reversible Lossless Paired Coordinates

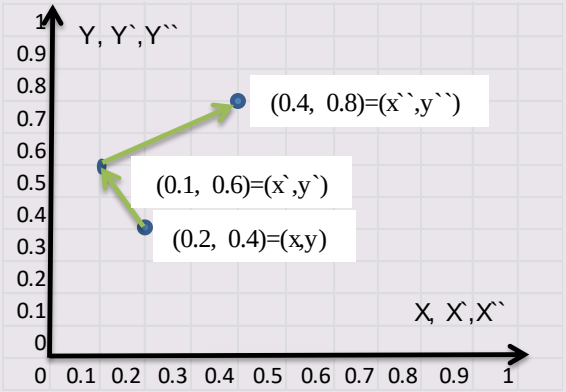


(a) Collocated Paired Coordinates.

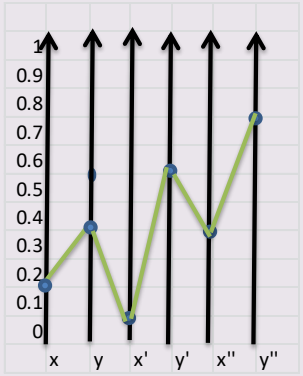


(b) Shifted Paired Coordinates.

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

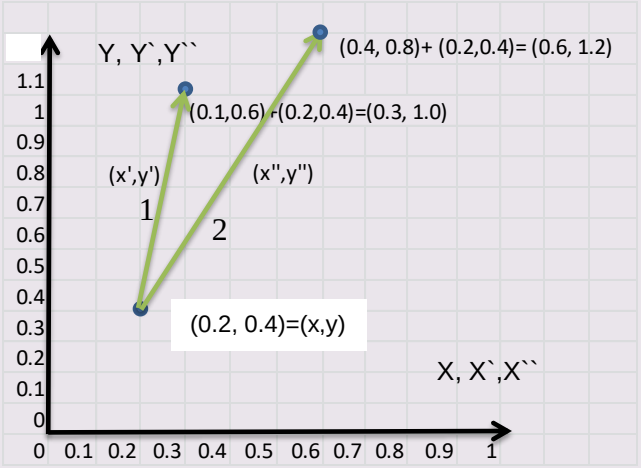


(a) Collocated Paired Coordinates



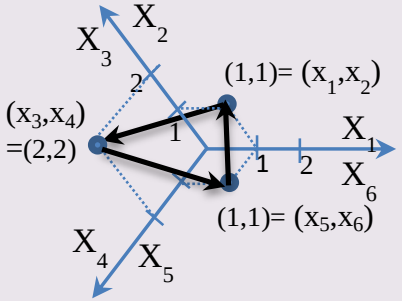
(b) Parallel Coordinates

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.

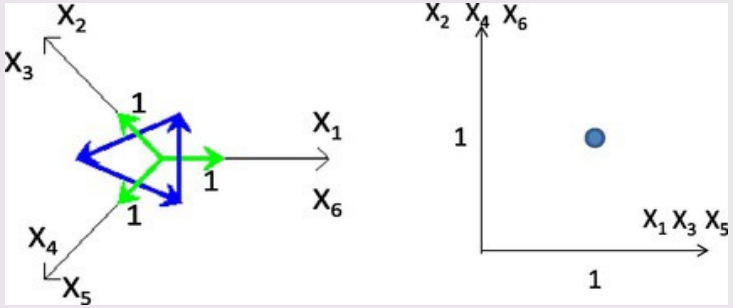


6-D point **x**=(x,y,x`,y`,x``,y``)=(0.2, 0.4, 0.1, 0.6, 0.4, 0.8) in Anchored Paired Coordinates with numbered arrows.

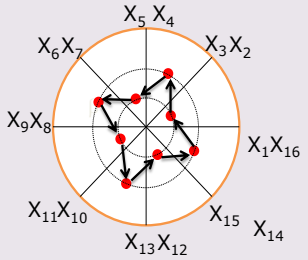
Reversible lossless Paired Coordinates

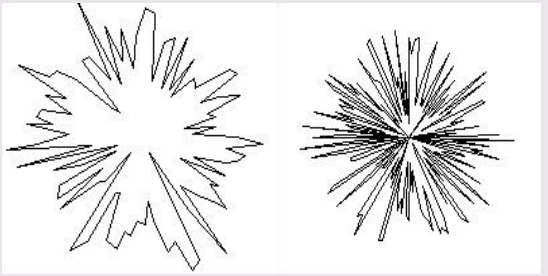


6-D point as a closed contour in 2-D where a 6-D point **x**=(1,1, 2,2,1,1) is forming a tringle from the edges of the graph in Paired Radial Coordinates with non-orthogonal Cartesian mapping.



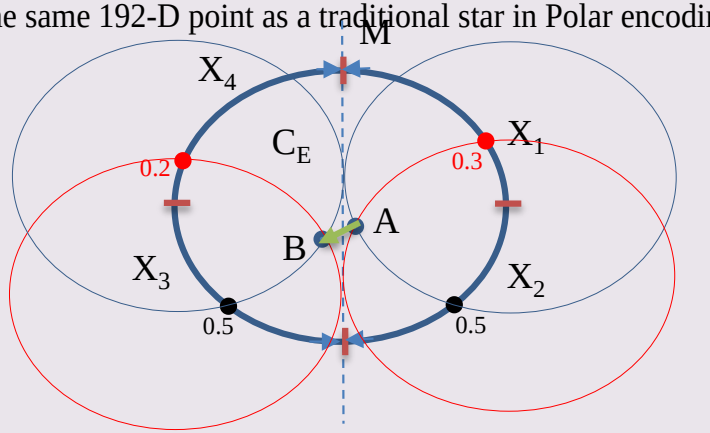
6-D point (1, 1, 1, 1, 1, 1) in two X1-X6 coordinate systems (left – in Radial Collocated Coordinates, right – in Cartesian Collocated Coordinates).



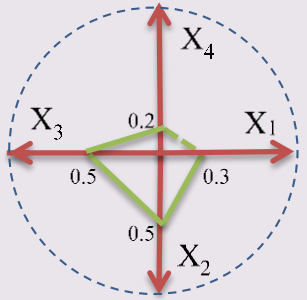


(a) (b) (c)

n-D points as closed contours in 2-D: (a) 16-D point (1,1,2,2,1,1,2, 2,1,1,2,2,1,1,2,2) 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.

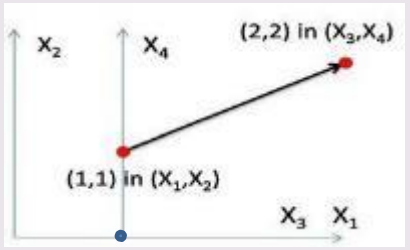


4-D point P=(0.3,0.5,0.5,0.2)) in 4-D Elliptic Paired Coordinates, EPC-H as a green arrow. Red marks separate coordinates in the Coordinate ellipse.

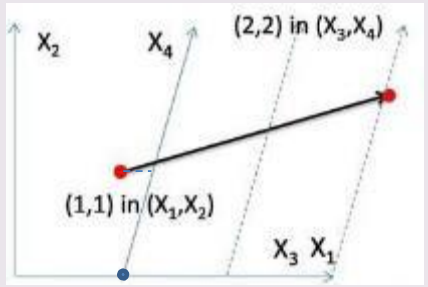


4-D point P=(0.3,0.5,0.5,0.2)) in Radial Coordinates.

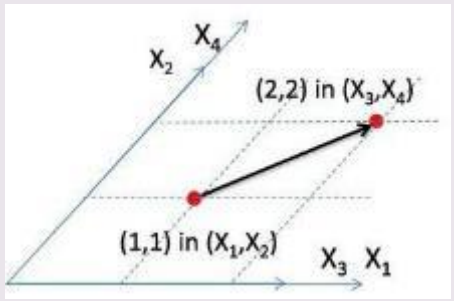
Reversible lossless Paired Coordinates



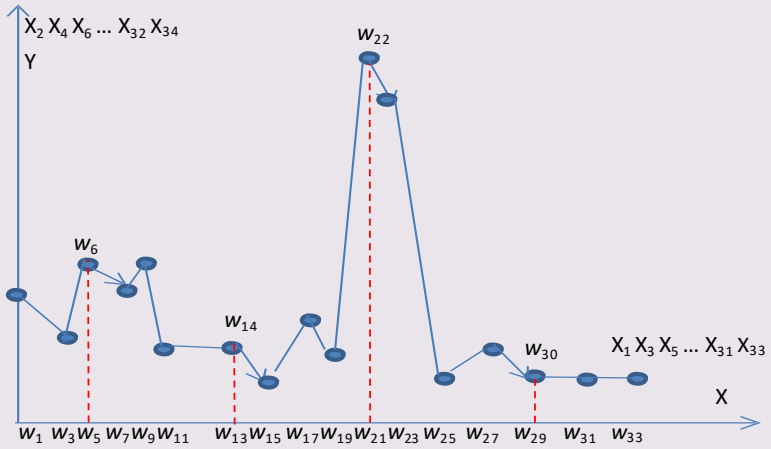
Partially Collocated Orthogonal (Ortho) Coordinates.



Partially Collocated Ortho and non-Ortho Coordinates.

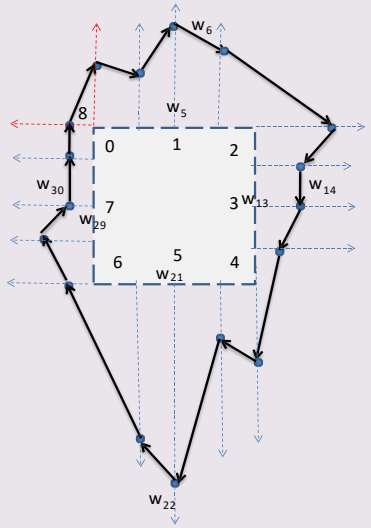


Collocated Paired non-Ortho Coordinates.

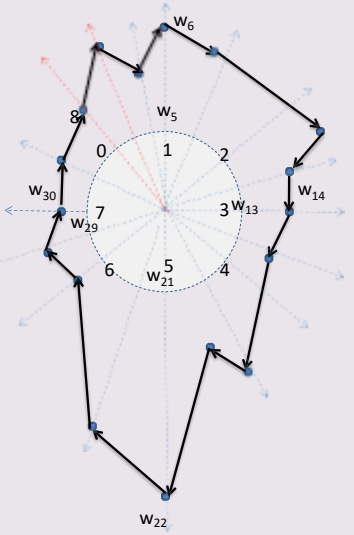


34-D point in Collocated Paired Coordinates.

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.



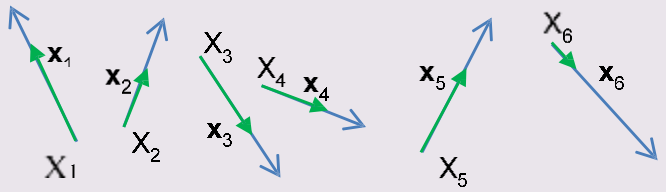
(a) Mapping odd coordinates to a square.



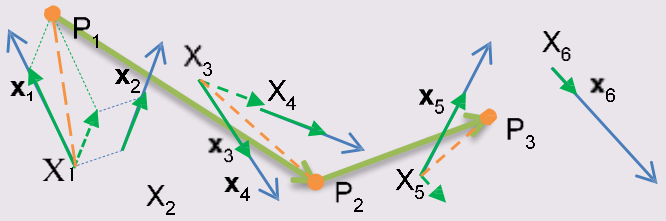
(b) Mapping odd coordinates to a circle.

34-D point from figure 2.32.in **Closed Paired Crown Coordinates** with odd coordinates mapped to the crown and even coordinates mapped dynamically to the normals to the crown.

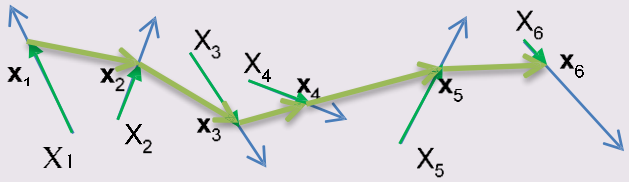
Graph construction algorithms in GLC



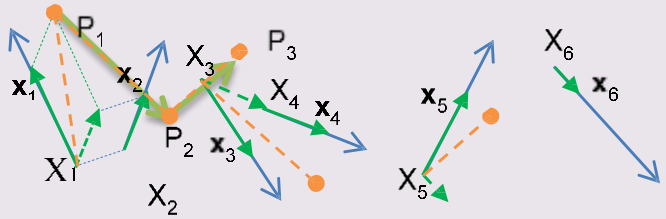
Six coordinates and six vectors that represent a 6-D data point (0.75,0.5,0.7,0.6,0.7, 0.3)



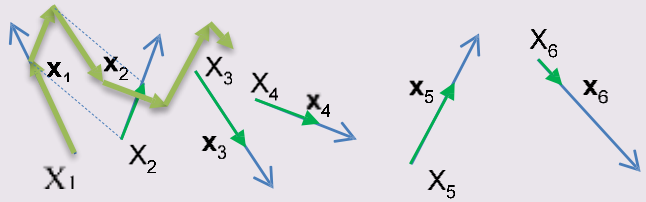
6-D data point (0.75,0.5,0.7,0.6,0.7, 0.3) in GLC-CC1



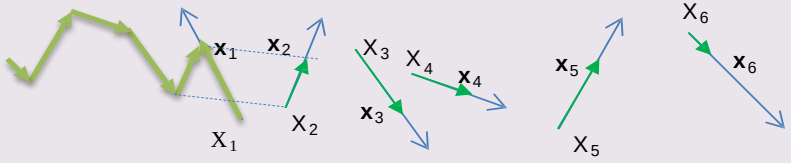
6-D data point (0.75,0.5,0.7,0.6,0.7, 0.3) in GLC-PC.



6-D data point (0.75,0.5,0.7,0.6,0.7, 0.3) in GLC-CC2



6-D data point (0.75,0.5,0.7,0.6,0.7, 0.3) in GLC-SC1.



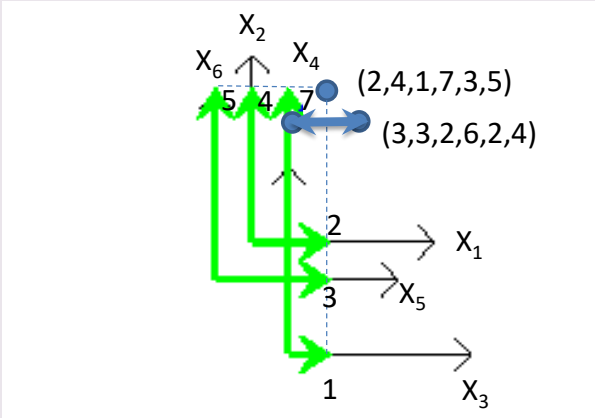
6-D data point (0.75,0.5,0.7,0.6,0.7, 0.3) in GLC-SC2

Math, theory and pattern simplification methodology: Statements

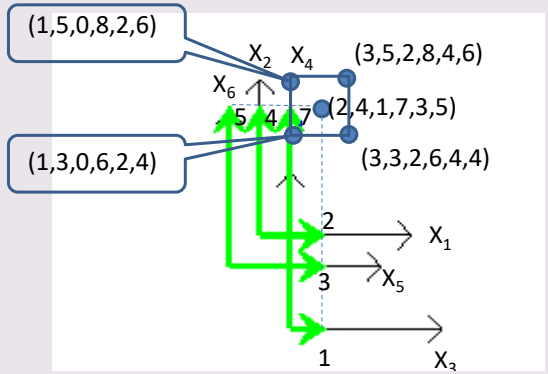
- **Statement 1.** Parallel Coordinates, CPC and SPC preserve L^p distances for p=1 and p=2, D(**x**,**y**) = D*(**x***,**y***).
- **Statement 2** (n points lossless representation). If all coordinates X_i do not overlap then GLC-PC algorithm provides bijective 1:1 mapping of any n-D point **x** to 2-D 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, $D(\mathbf{x},\mathbf{y}) = D^*(\mathbf{x}^*,\mathbf{y}^*)$.
- **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 $\mathbf{x}=(x_1, x_2,,..,x_n)$ 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

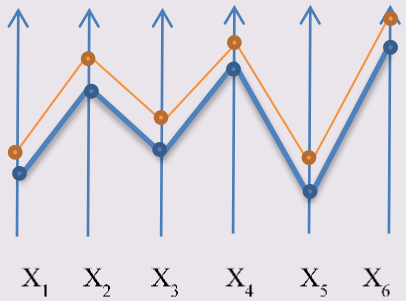


6-D points (3,3,2,6,2,4) and (2,4,1,7,3,5) in X_1-X_6 coordinate system build using point (2,4,1,7,3,5) as an anchor.

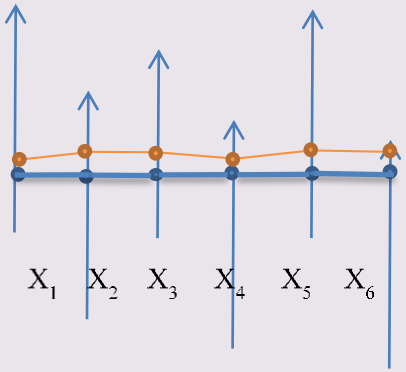


Data in Parameterized Shifted Paired Coordinates. Blue dots are corners of the square S that contains all graphs of all n-D points of hypercube *H* for 6-D base point (2,4,1,7,3,5) with distance 1 from this base point.

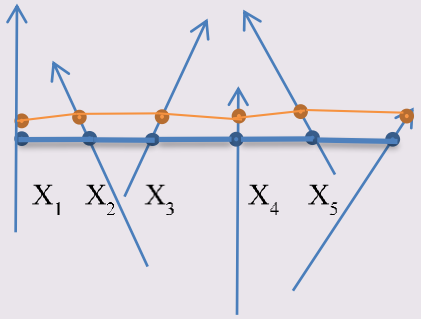
Adjustable GLCs for decreasing occlusion and pattern simplification



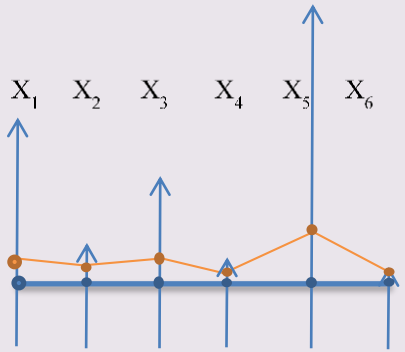
(a) Data in Parallel Coordinates – non-preattentive representation.



(b) Data in the Shifted Parallel Coordinates - preattentive representation.



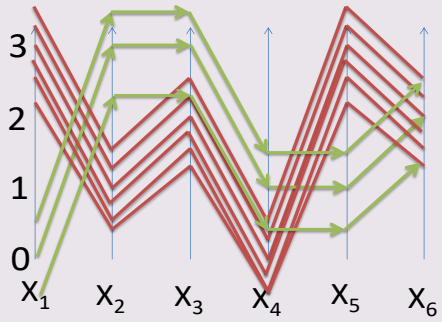
(c) Data in the Shifted General Line Coordinates- preattentive representation.



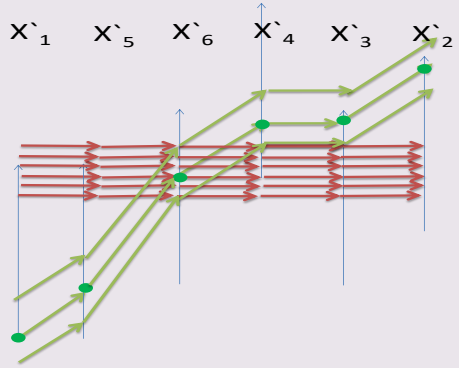
(d) Data in the scaled Parallel Coordinates – preattentive representation

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



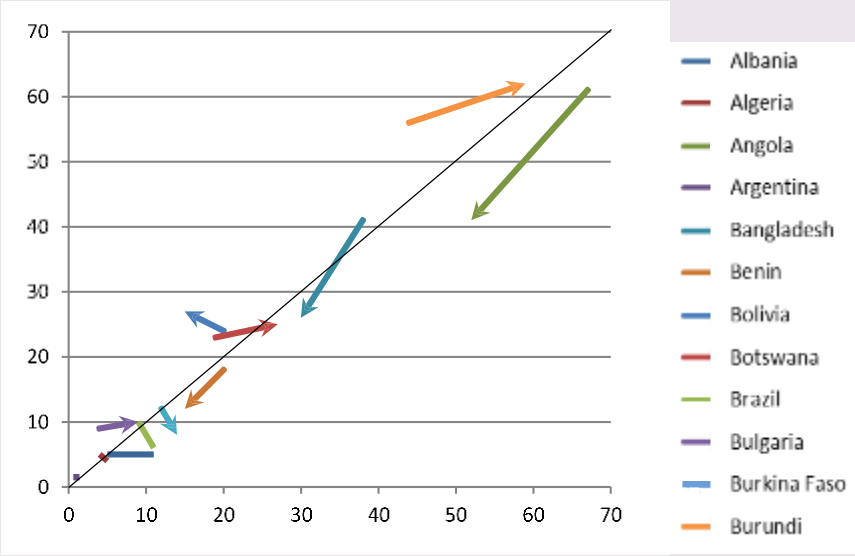
(a) Original visual representation of the two classes in the Parallel Coordinates,



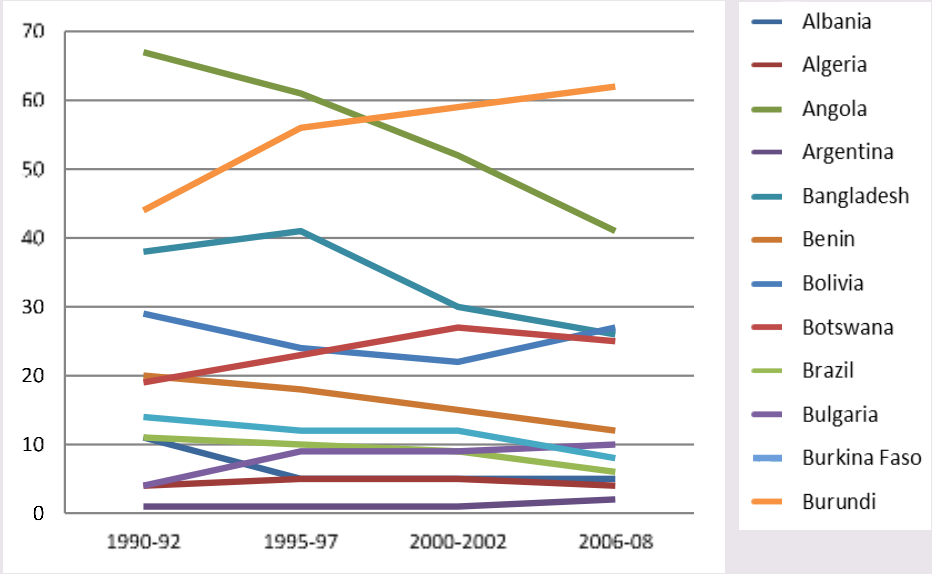
(b) Simplified visual representation after the shifting and reordering of the Parallel Coordinates.

Simplification of the visual representation by the shifting and reordering of the Parallel Coordinates

Case Studies: World Hunger data



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



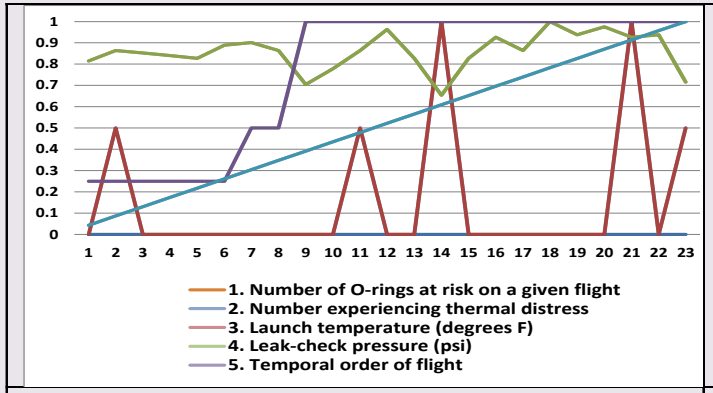
4-D data: representation of prevalence of undernourished in the population (%) in traditional time series (equivalent to Parallel Coordinates for time series)

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

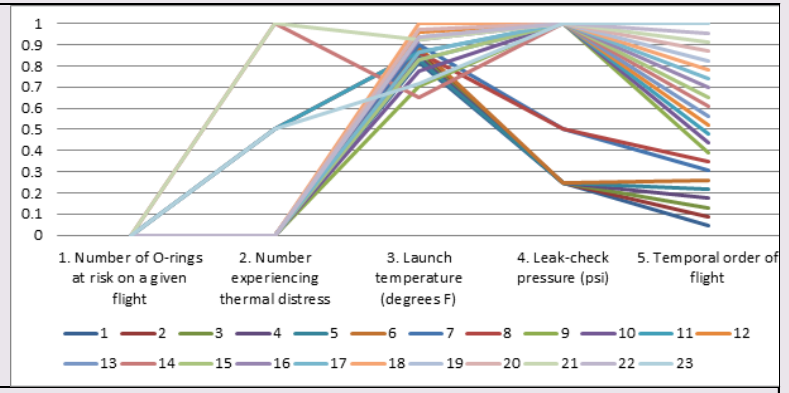
$$GHI = (UNN+UW5+MR5)/3,$$

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 %).

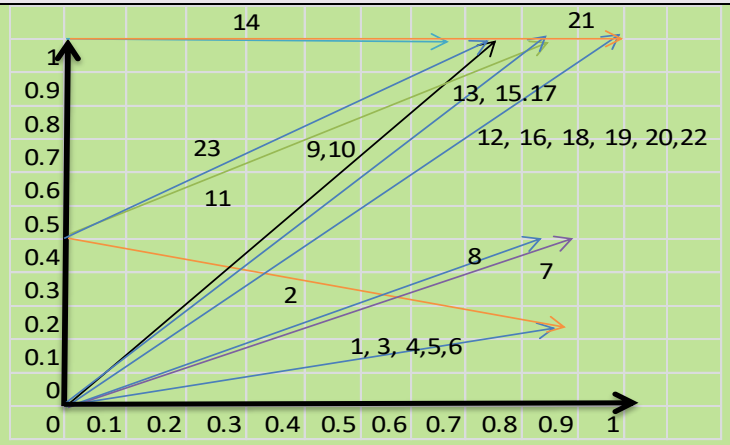
Challenger Space Shuttle data



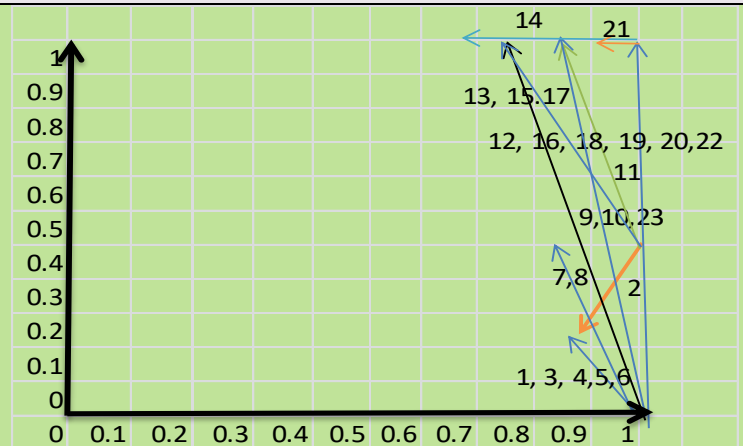
Challenger USA Space Shuttle normalized O-Ring dataset in a traditional line chart. X coordinate is a temporal order of flights. Each line represents an attribute.



Challenger USA Space Shuttle normalized O-Ring dataset in a traditional line chart. X coordinate is a temporal order of flights. Each line represents a flight.



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.

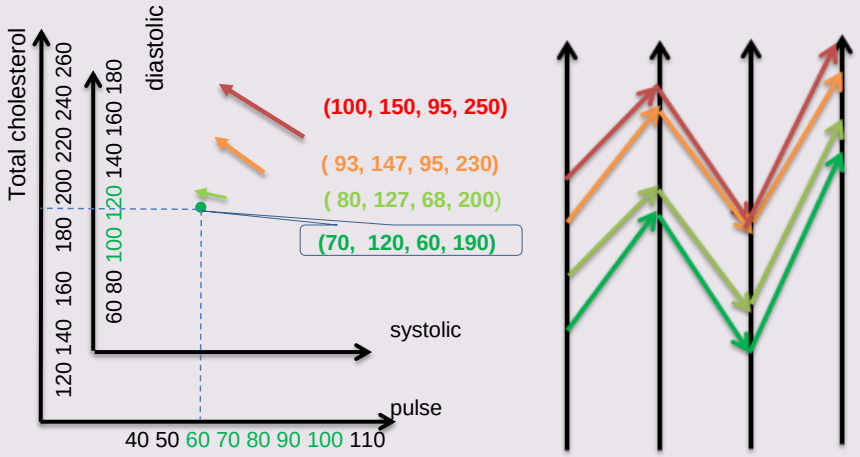


Challenger USA Space Shuttle normalized O-Ring dataset in the Collocated Paired Coordinates. X and Y coordinates are normalized values of attributes. 6 O-rings at risk are normalized as 1.

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].

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 (53F) 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



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.

(b) the same data as in (a) in Parallel Coordinates.

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.

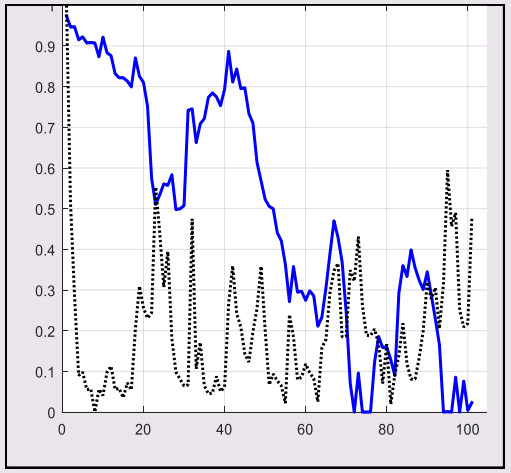
The colors show the progress to the goal.

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

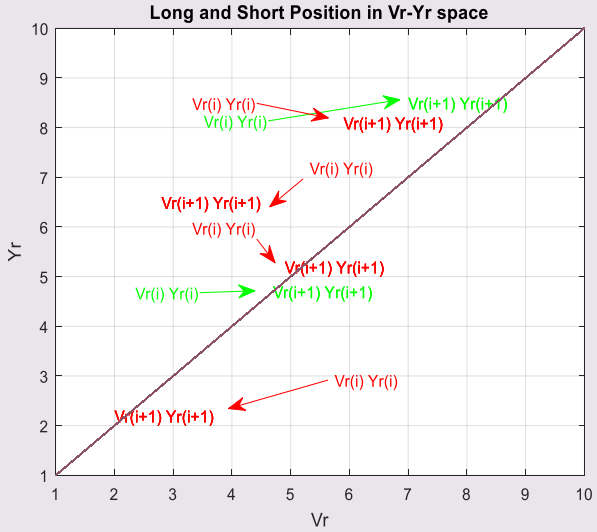
- Experiments -- people quickly grasp how to use this health monitor.
- This health monitor is expandable.
- Two more indicators is another pair of shifted Cartesian Coordinates.
  - *The goal is the same dark green 2-D dot*
  - *Each graph has two connected arrows.*
- Graphs closer to the goal are smaller.

Case studies: Knowledge Discovery and Machine Learning for Investment Strategy with CPC

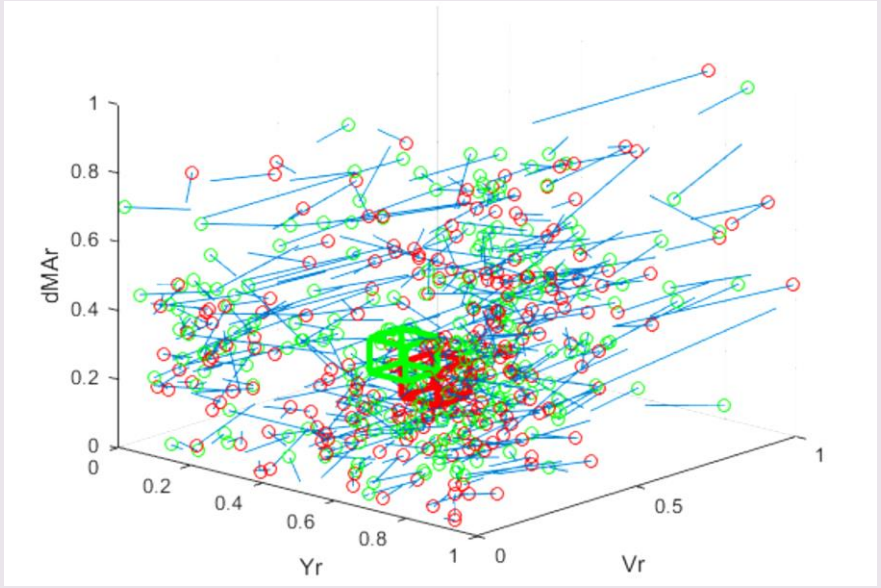
- The CPC visualization shows **arrows** in (V_r,Y_r) space of **volume V_r** and **relative main outcome variable Y_r**
- This is a part of the data shown as traditional time series with *time* axis.
- CPC has no time axis. The **arrow direction shows time**.
- The arrow beginning is the point in the space $(V_{r\,i},Y_{r\,i})$, and its head is the **next time point** in the collocated space $(V_{r\,i+1},Y_{r\,i+1})$.
- CPC give 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 $Y_{r\,i+1}>Y_{r\,i}$ then $Y_{i+1}>Y_i$ 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 indicates **small** profit*
  - *A **more vertical** arrows indicates the **larger profit**.*
- In comparison with traditional time series, the CPC bring the **additional knowledge about the potential of profit** in selected area of parameters in (V_r,Y_r) space.



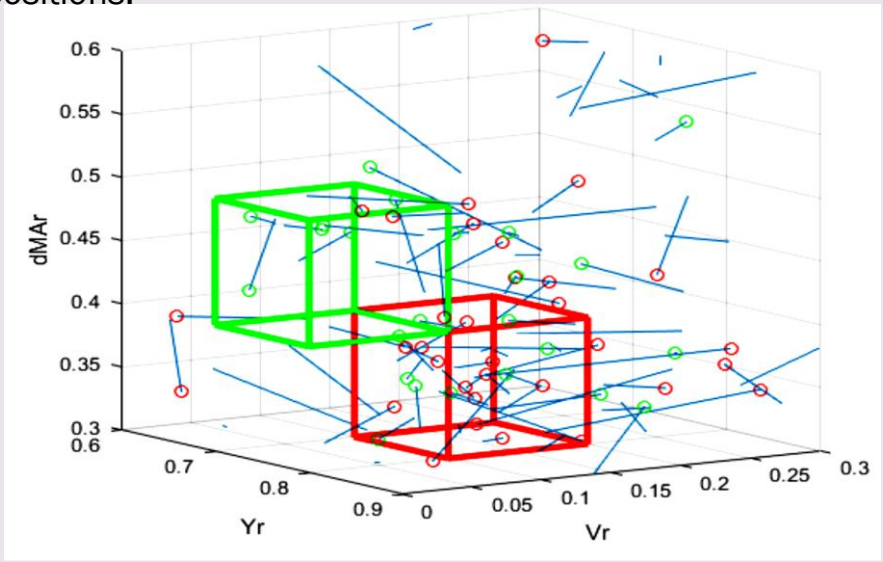
Comparison of two time series: relative outcome *Yr* and relative volume in every one hundred period.



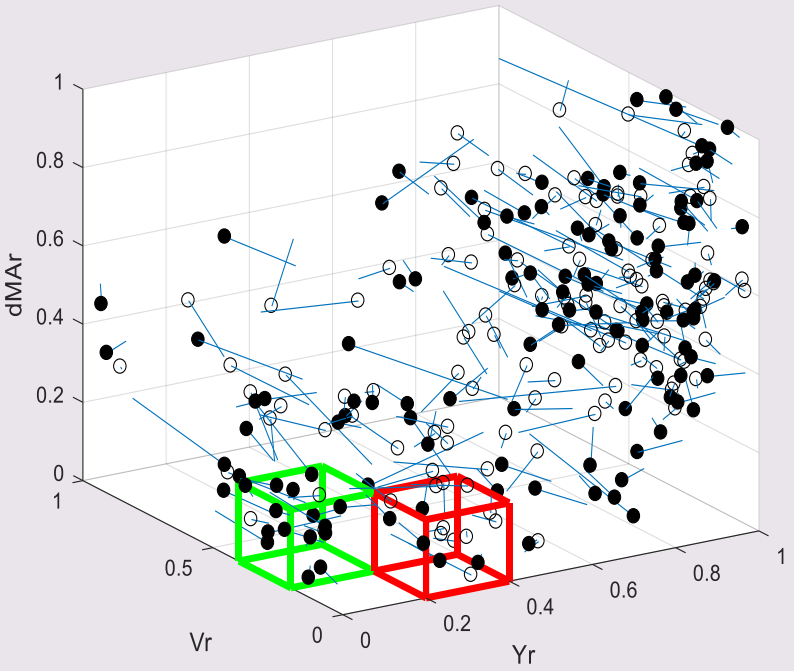
Some examples of arrows which show points of two time series, Y_r and V_r



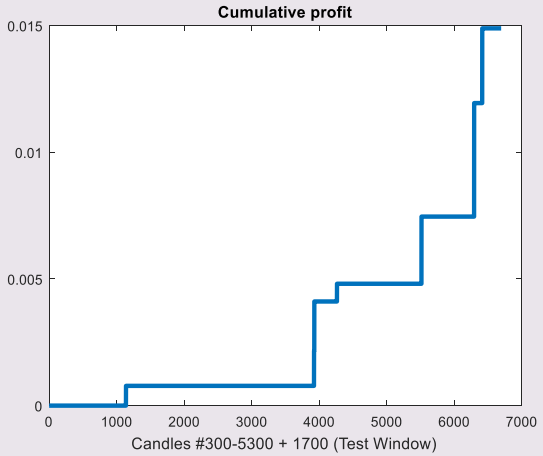
Pins in 3-D space: two cubes found in (Yr,dMAr,Vr) space with the maximum asymmetry between long and short positions.

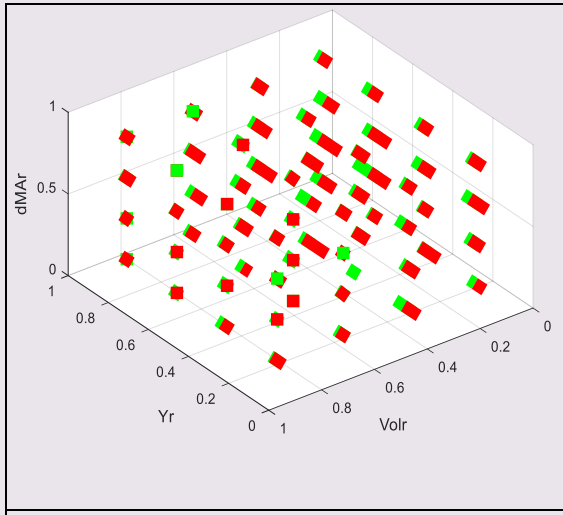


The zoomed cubes with the best asymmetry from Figure 8.16. The upper cube with green circles is selected for long positions lower cube with red

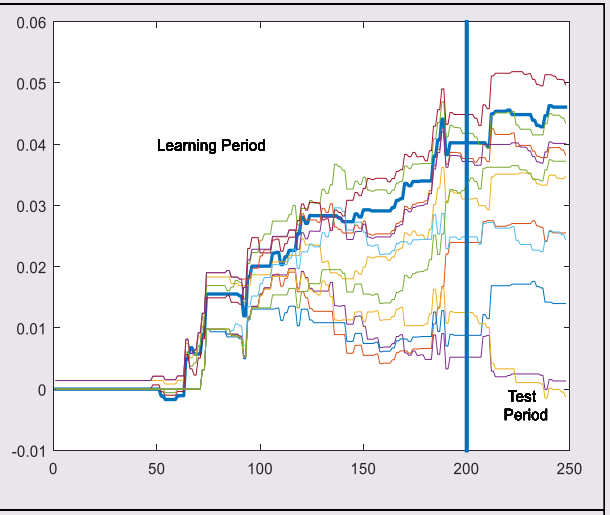


Two determined cubes in Y_r-dMA_r-V_r space with the maximum asymmetry between long and short positions for the new grid resolution.



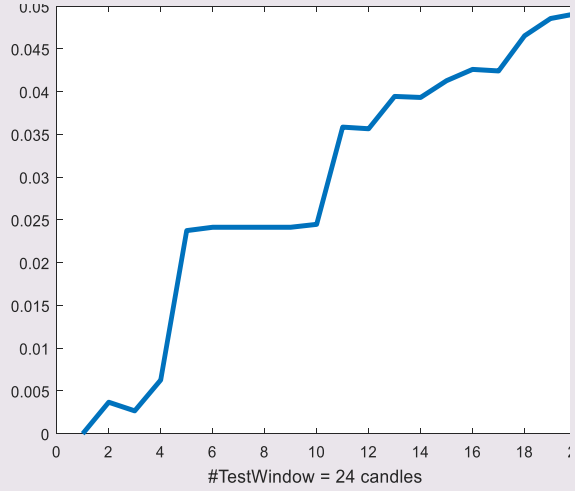


(a) Bars in 3D space after learning period which represent preferences to open long (green) or short (red) position.

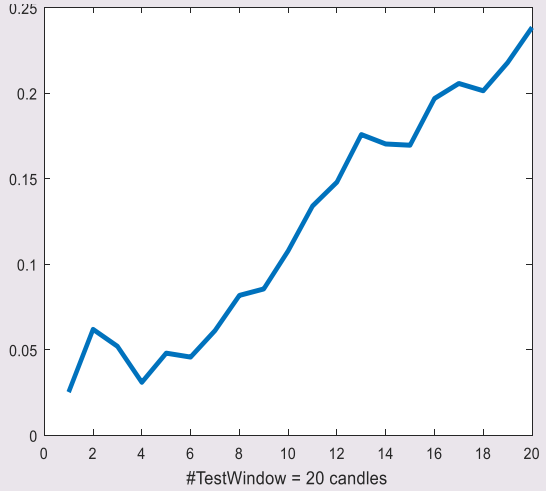


(b) Cumulative profits for different values of parameters (f,s,k_b).

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).



Cumulative profit for main strategy with learning windows of 100 1h-candles and test widows of 24 1h-candles
(Calmar ratio ~18, PPC=0.87).



Cumulative profit for the strategy in 1d EURUSD time series with 5-days testing windows and 30-days learning windows in 20 periods, Calmar ratio=7.69; PPC=23.83 pips; delta=0.05.

- 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 = (average of return over time Δt)/max drawdown over time Δt, 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*.

Time	*Description*
t_1	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.
t_2	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.
t_3	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.
t_4	Agent *G* retrieves semantic features (words, phrases) $F(P_2)$ of P_2.
t_5	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*).
t_6	Agent *G* reevaluate usuality of P_2 taking in to account these correlations and concludes that P_2 is unusual.

Confusion matrix for visual classification rule R1

	Predicted joke	Predicted non-joke
Actual jokes 17	15	2
Actual non-jokes 17	2	15

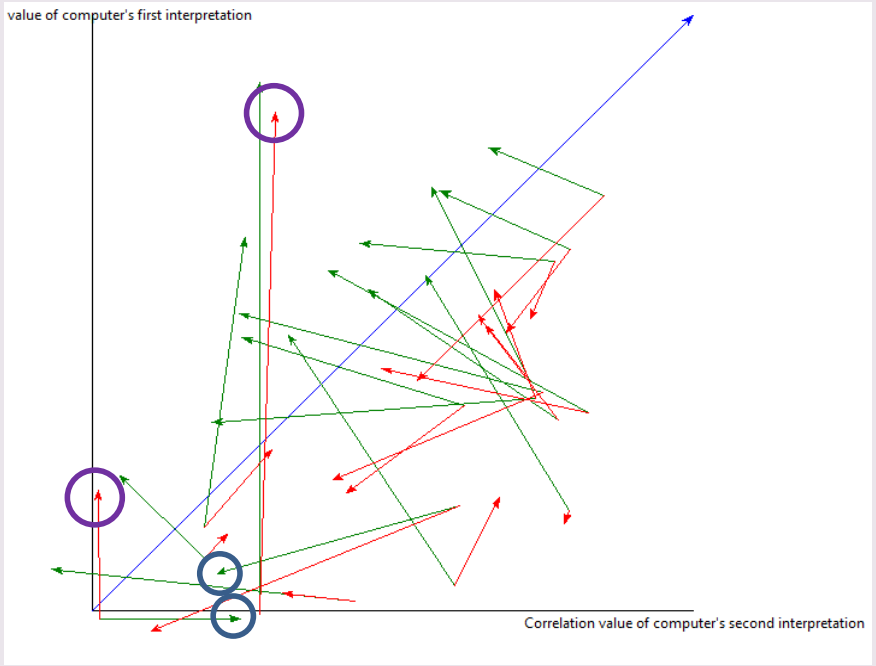
We record this visual discovery as a rule R2:

R2: If $z_3 < z_4$ then **z** is a joke, else **z** is a non-joke.

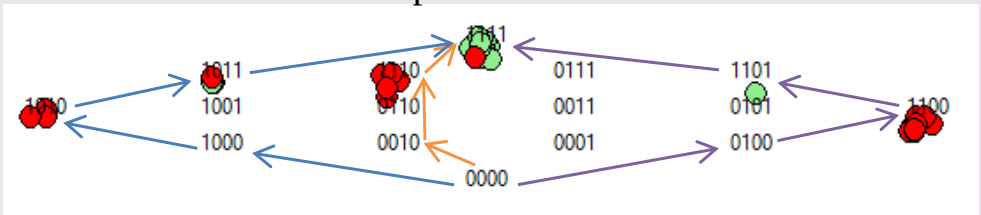
The resulting C4.5 rule R3 is

If $z_3 < z_4 < 0.0075$ then **z** is a joke,

else **z** is a non-joke.



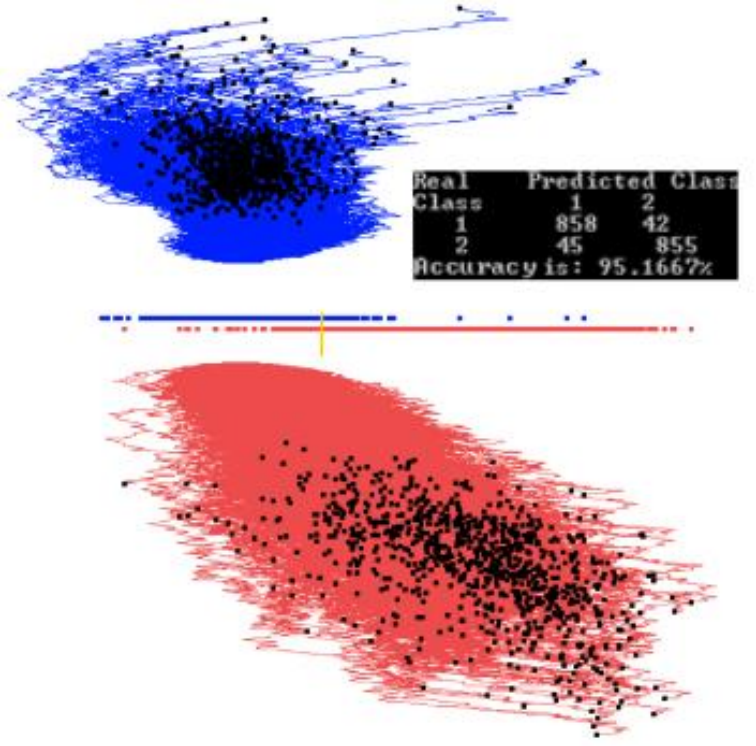
Collocated Paired Coordinate plot of meaning context correlation over time. The set of jokes and non-jokes plotted as meaning correlation over time using collocated paired coordinates.



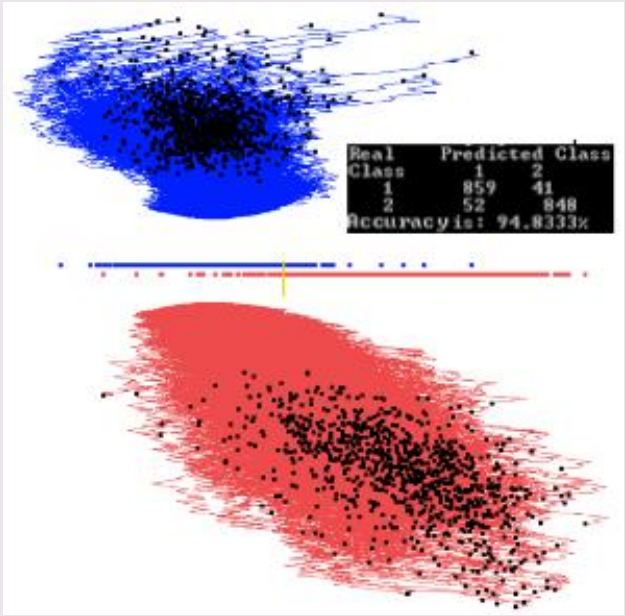
Monotone Boolean space of jokes (green dots) and non-jokes (red dots).

R4: If ($\mathbf{w} \geq (1,0,1,1) \vee \mathbf{w} \geq (0,1,0,1)$ then **w** is joke else **w** is a non-joke.

Case study: Recognition of digits Dimension reduction



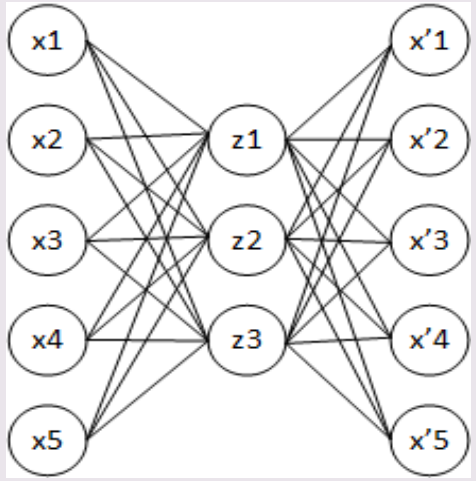
Experiments with 900 samples of MNIST dataset for digits 0 and 1.Results for the best linear discriminant function of the first run of 20 epochs.



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%.

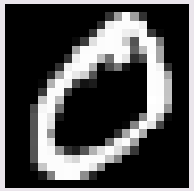
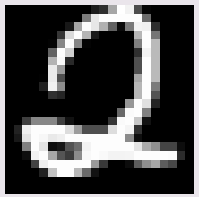
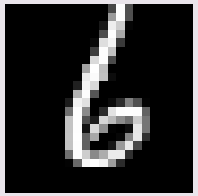
Case study: Recognition of digits
Dimension reduction

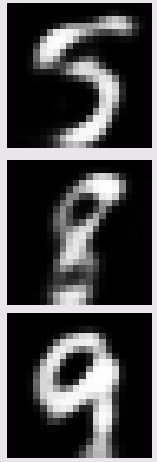
Lossy but controlled Dimension Reduction with NN autoencoder



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

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

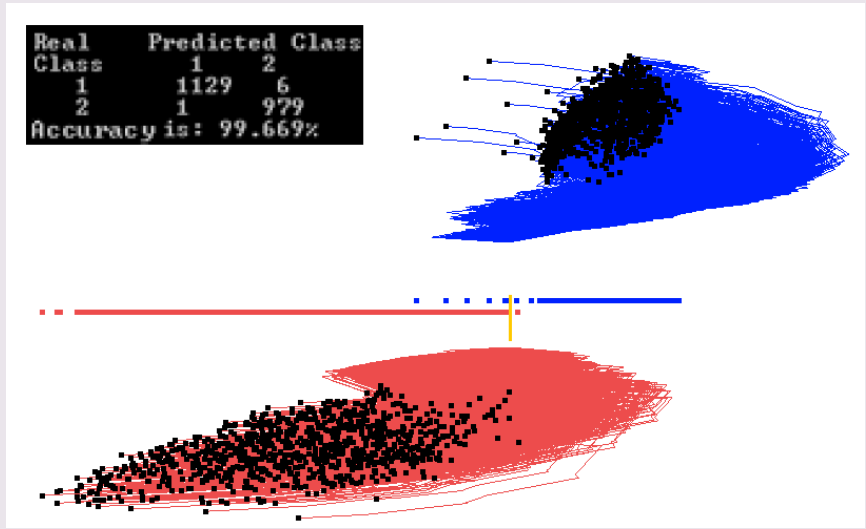






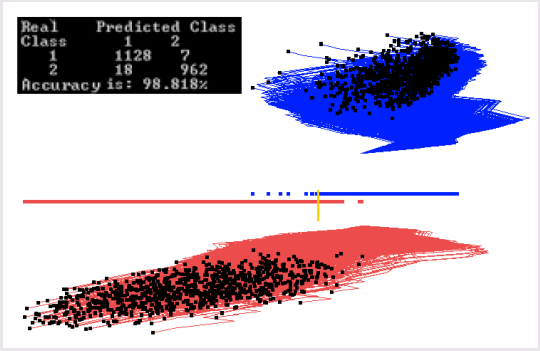
Decoded images with hidden Layer: 24 nodes

- 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

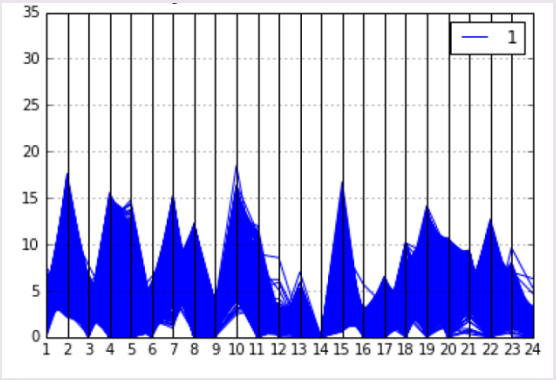
Case study: Digit recognition, Dimension reduction



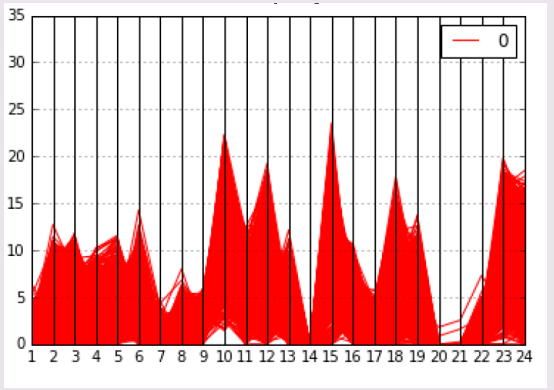
Encoded digit 0 and digit 1 on and GLC-L, using 24 dimensions found by the Autoencoder among 484 dimensions. Results for the best linear discriminant function of the first run of 20 epochs.



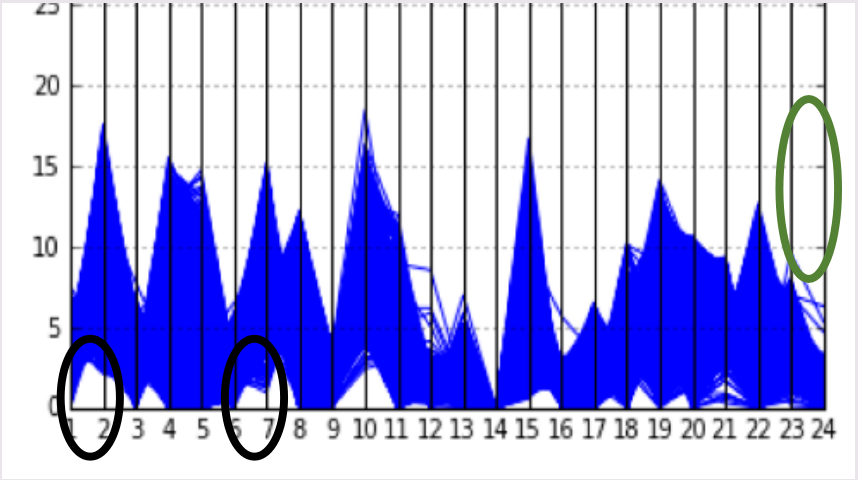
Encoded digit 0 and digit 1 on GLC-L, using 24 dimensions found by the Autoencoder among 484 dimensions. Another run of 20 epochs, best linear discriminant function from this run. Accuracy drops 1%.



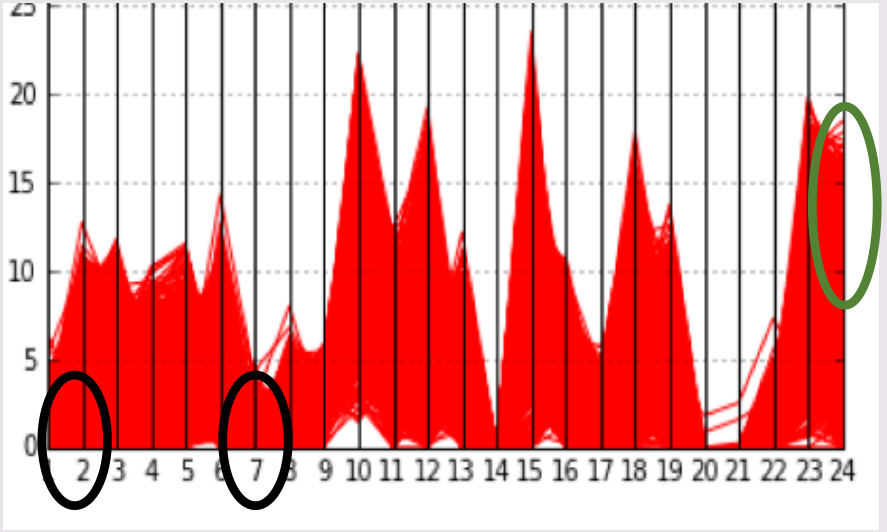
Encoded digit 1 in Parallel Coordinates using 24 dimensions found by the Autoencoder among 484 dimensions. Each vertical line is one of the 24 features scaled in the [0,35] interval. Digit 1 is visualized on the parallel coordinates.



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.



(a) digit "1"



(**b**) digit "0"

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.

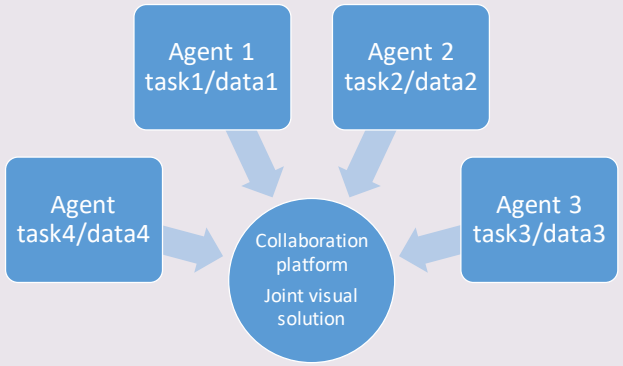
Visual rule design from embedding generated by auto-encoder

- 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.,*

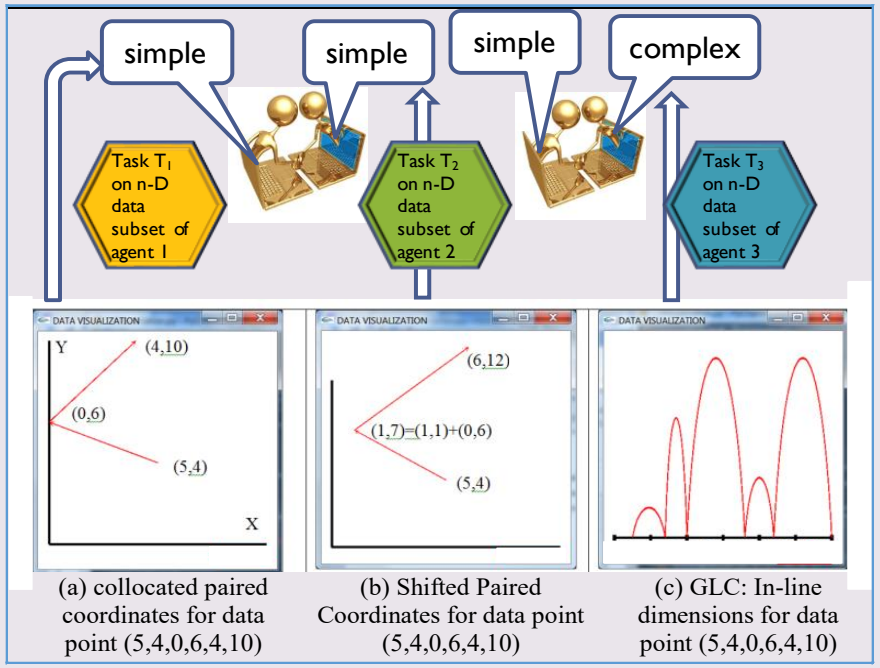
    if x_2=0 then 0; if x_7=0 then 0.

    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.*

Collaborative visual discovery



Data and Task Split-based Collaborative Visualization framework.



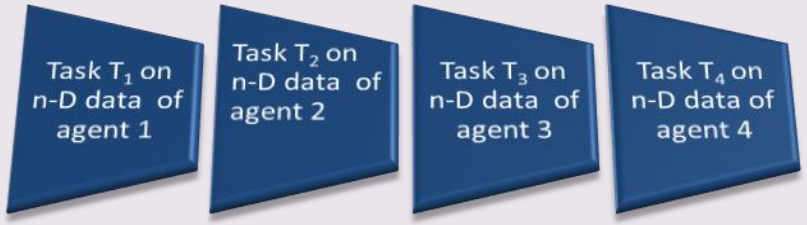
(a) collocated paired coordinates for data point (5,4,0,6,4,10)

(b) Shifted Paired Coordinates for data point (5,4,0,6,4,10)

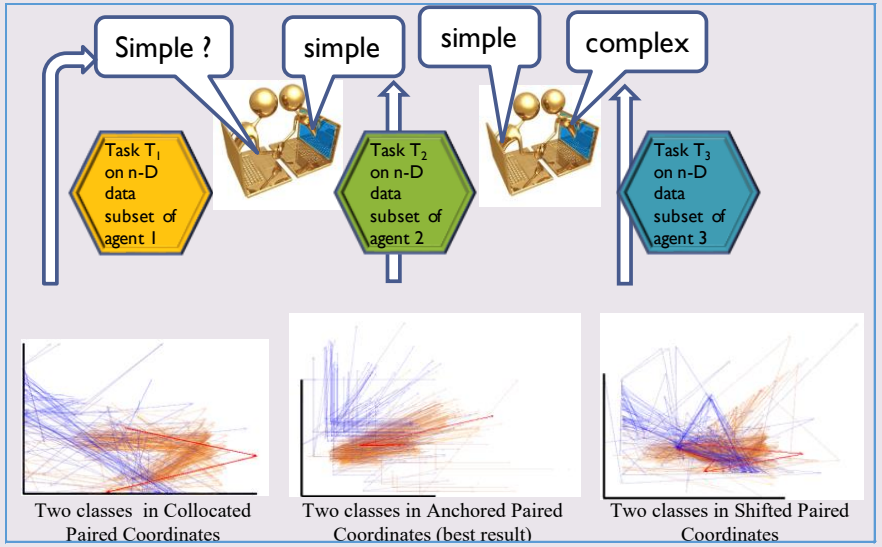
(c) GLC: In-line dimensions for data point (5,4,0,6,4,10)

Collaboration diagram with data example 1.

Data splitting for Collaborative Visualization.



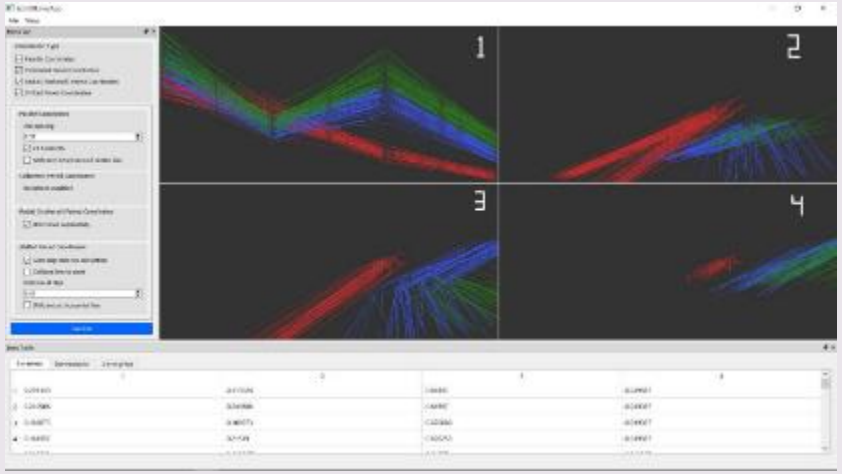
Task splitting for Collaborative Visualization.

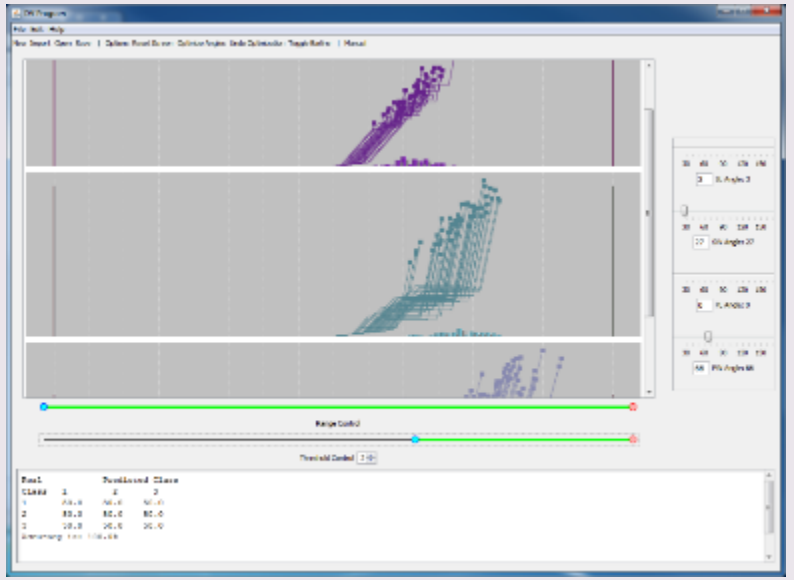


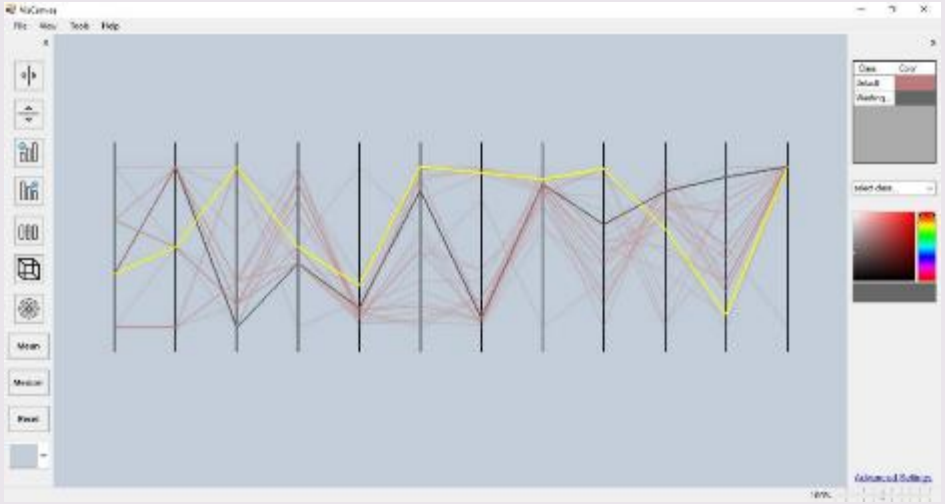
Collaboration diagram with data example 2.

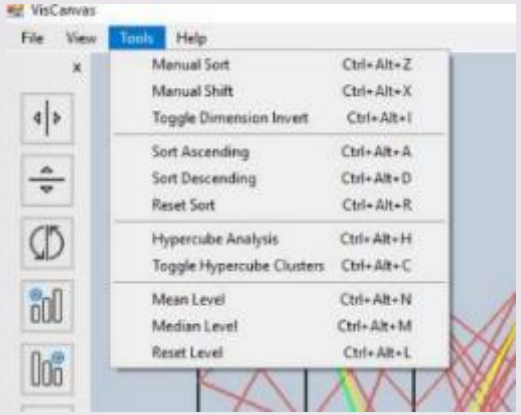
Experiments: success in dimension n>100

Current Software



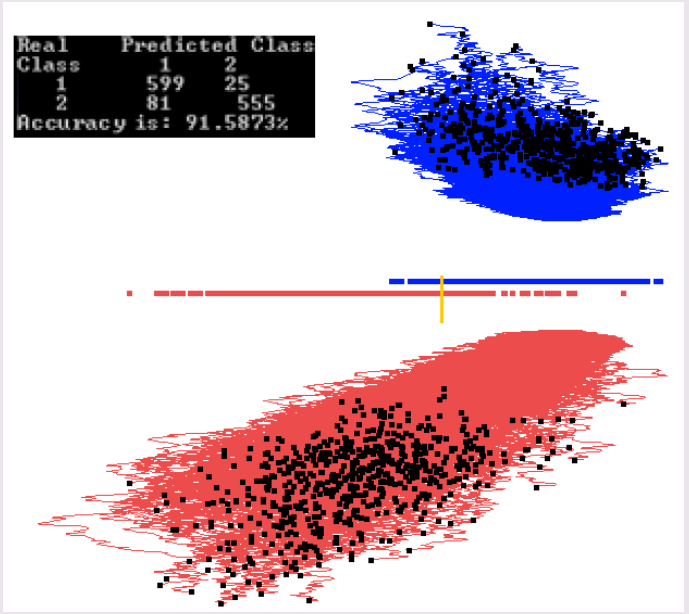




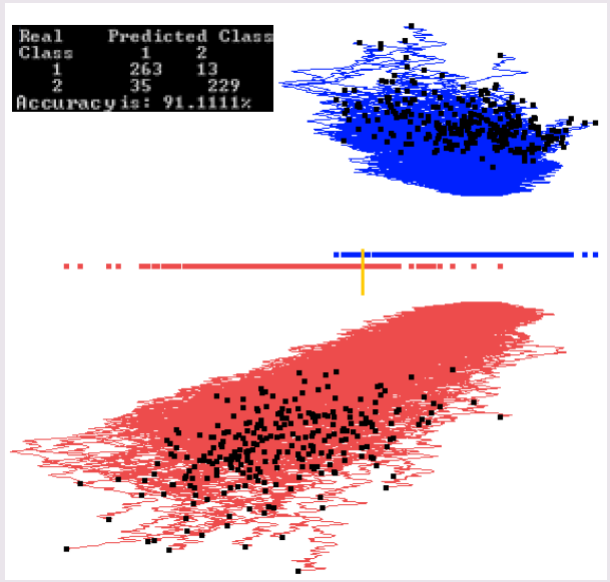


Case study: Recognition of digits

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.

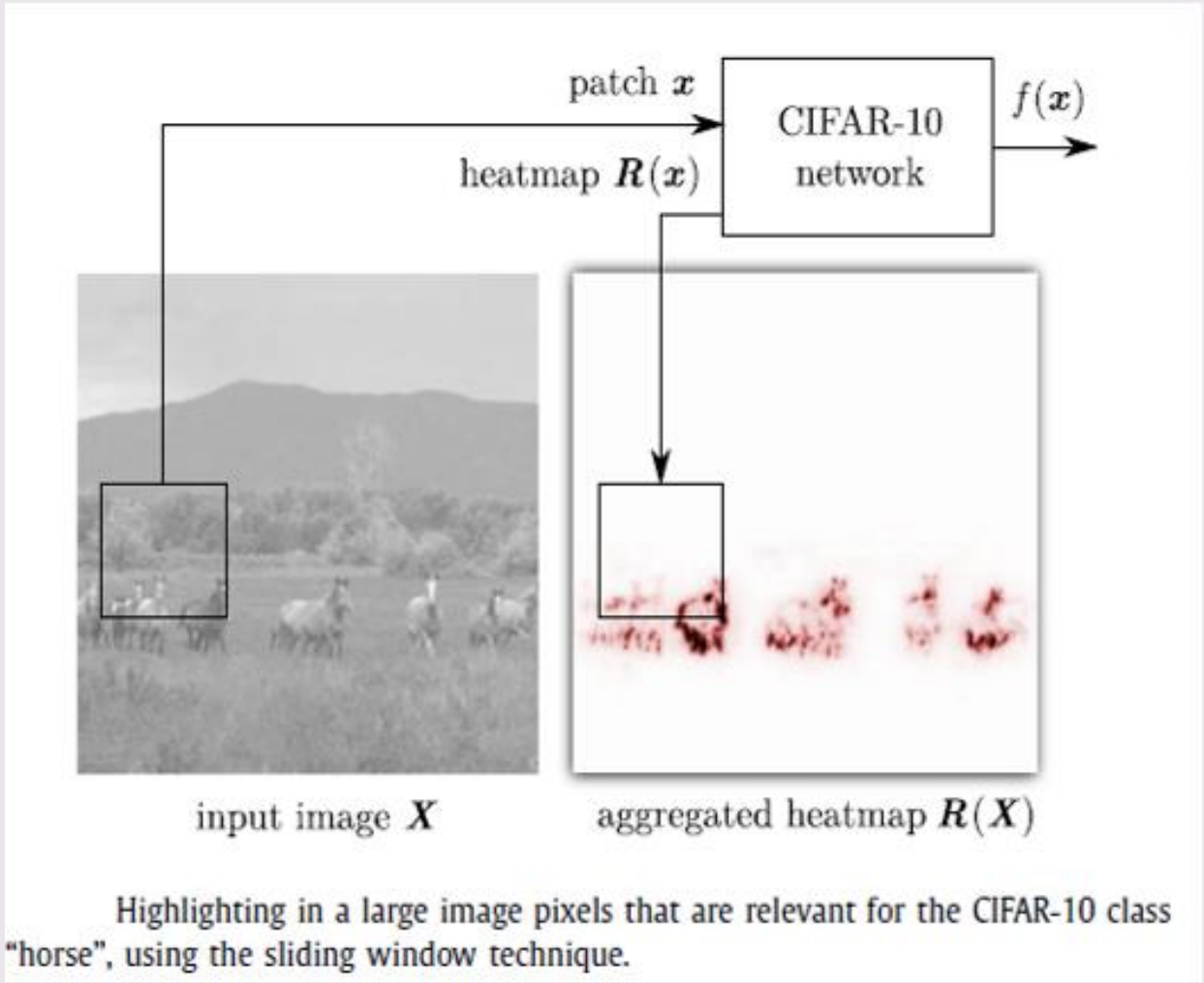
Approaches to explain deep and other ML models by visual means

- 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**

Visualization of Image features in heat map



Highlighting in a large image pixels that are relevant for the CIFAR-10 class "horse", using the sliding window technique.

CIFAR-10 classification benchmark problem is to classify RGB 32x32 pixel images across 10 categories

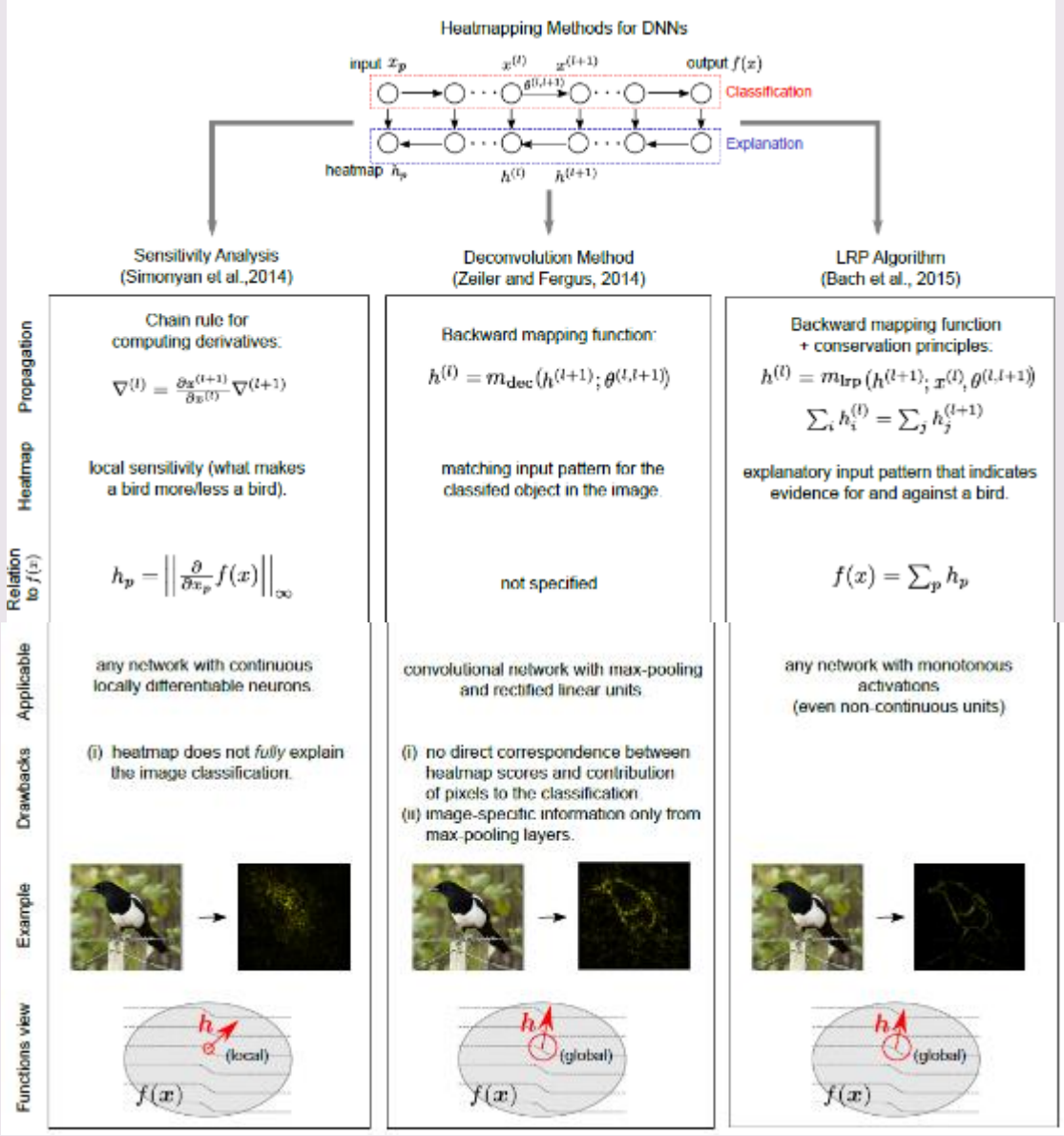
- Tensor flow CIFAR-10 CNN multi-layer network with
  - *alternating convolutions and nonlinearities followed by fully connected layers and softmax classifier.*
  - *Peak accuracy ~ 86%*
  - *few hours of training on a GPU.*
  - *~ 1M learnable parameters*
  - *19.5M multiply-add operations to compute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

Montavon G, Samek W, Müller KR. Methods for interpreting and understanding deep neural networks. Digital Signal Processing. 2018 Feb 1;73:1-5.

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.*

Samek W, Binder A, Montavon G, Lapuschkin S, Müller KR. Evaluating the visualization of what a deep neural network has learned. IEEE transactions on neural networks and learning systems. 2017 Nov;28(11):2660-73.



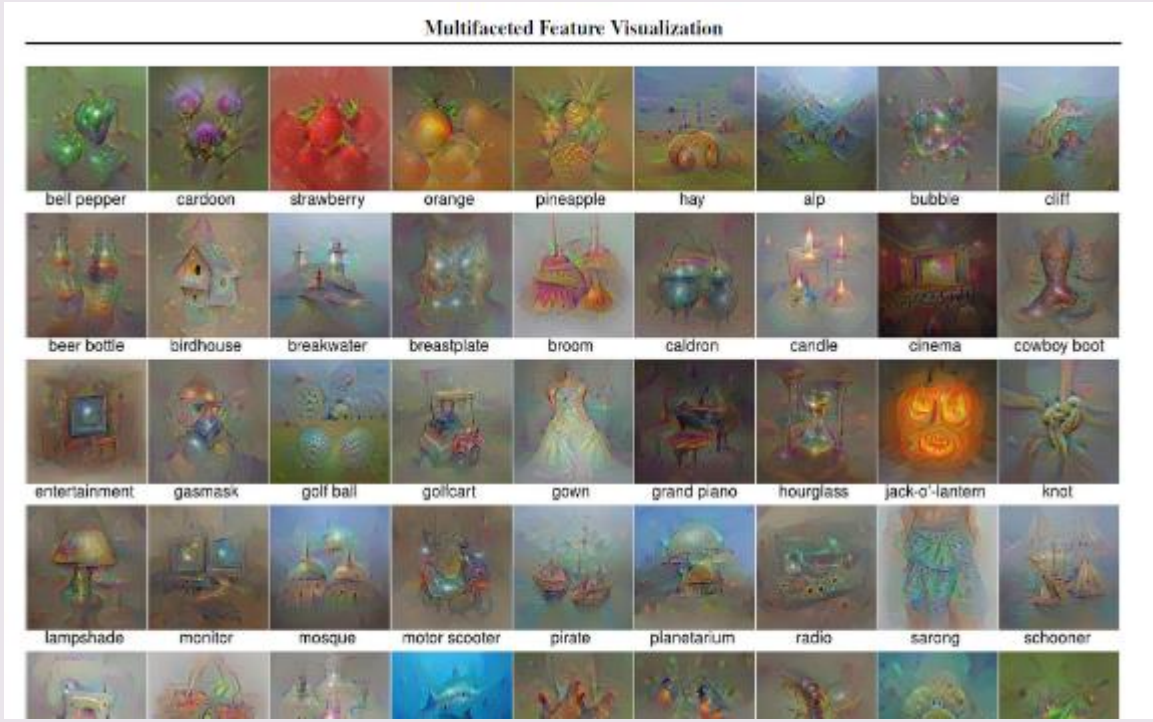
- Comparison of the three heatmap computations
- Left:
  - ***Sensitivity heatmaps*** *(local explanations) -- measure* ***change of the class*** *when specific pixels are changed based on partial* ***derivatives.***
  - *applicable to architectures with differentiable units.*
- Middle:
  - ***Deconvolution 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.*

Samek W, Binder A, Montavon G, Lapuschkin S, Müller KR. Evaluating the visualization of what a deep neural network has learned. IEEE transactions on neural networks and learning systems. 2017 Nov;28(11):2660-73.

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**.*

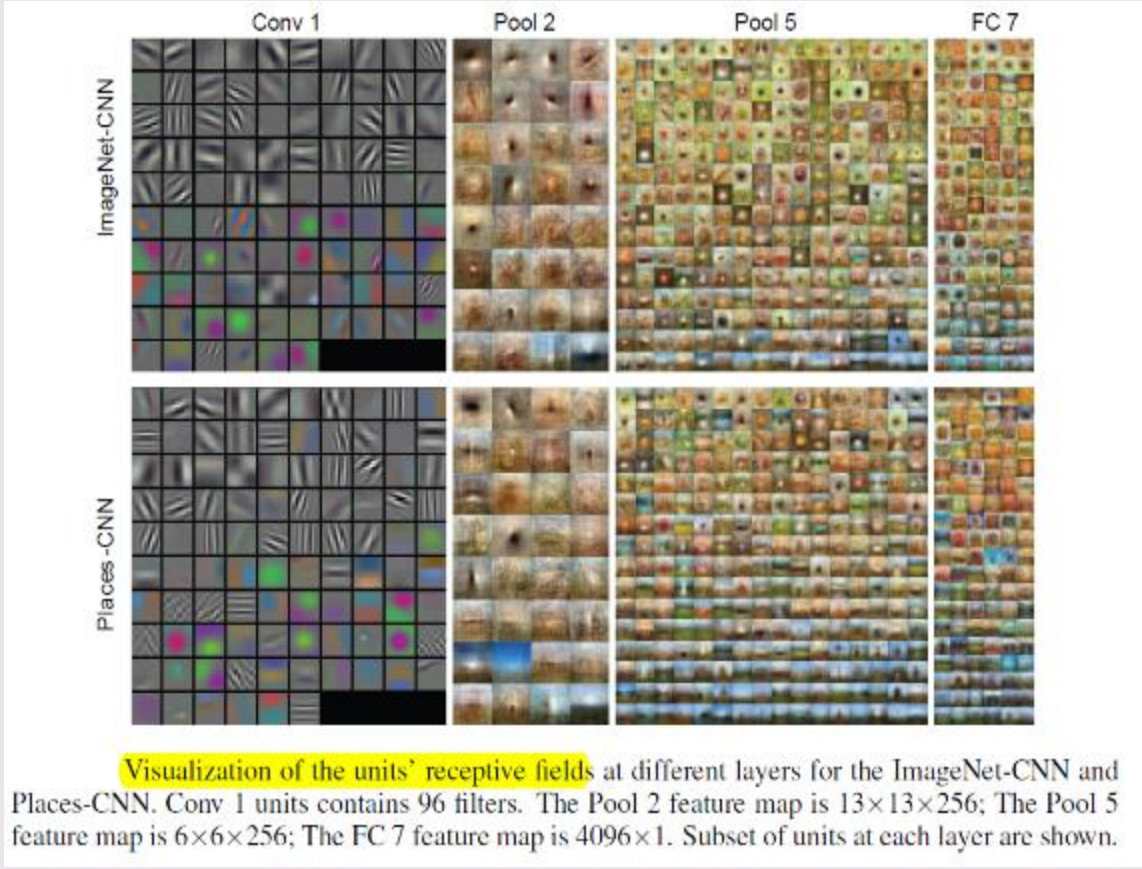


Nguyen, A., Yosinski, J., Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In Proc. of the CVPR Conference, 2015.

Nguyen A, Yosinski J, Clune J. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. arXiv preprint arXiv:1602.03616. 2016.

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.*

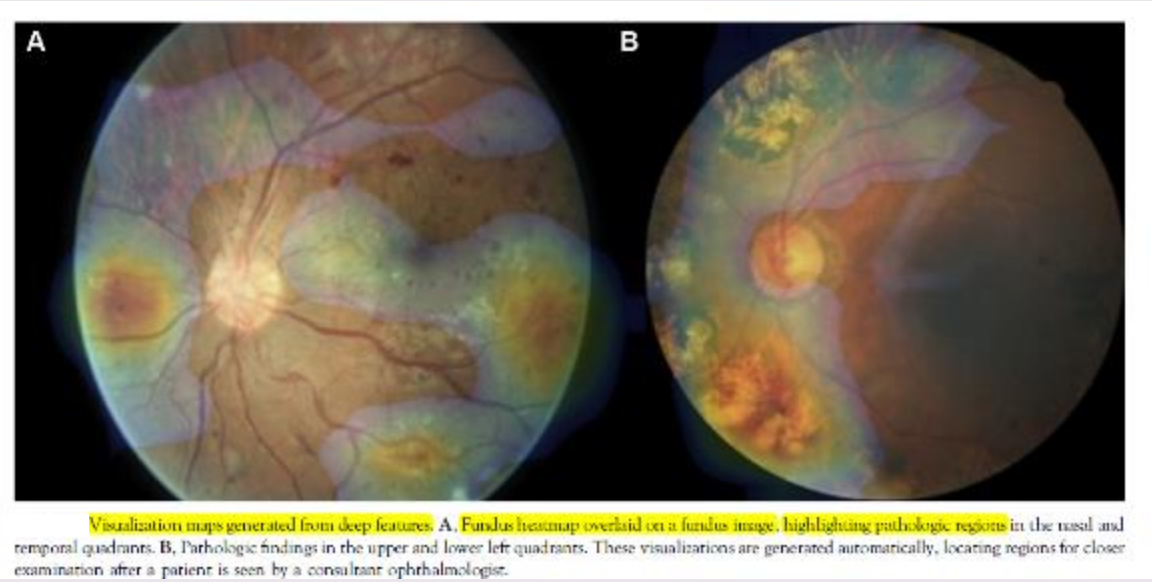


Visualization of the units' receptive fields at different layers for the ImageNet-CNN and Places-CNN. Conv 1 units contains 96 filters. The Pool 2 feature map is 13×13×256; The Pool 5 feature map is 6×6×256; The FC 7 feature map is 4096×1. Subset of units at each layer are shown.

Zhou B, Lapedriza A, Xiao J, Torralba A, Oliva A. Learning deep features for scene recognition using places database. In: Advances in neural information processing systems 2014 (pp. 487-495).

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.



Visualization maps generated from deep features. A, Fundus heatmap overlaid on a fundus image, highlighting pathologic regions in the nasal and temporal quadrants. B, Pathologic findings in the upper and lower left quadrants. These visualizations are generated automatically, locating regions for closer examination after a patient is seen by a consultant ophthalmologist.

Gargeya R, Leng T. Automated identification of diabetic retinopathy using deep learning. Ophthalmology. 2017 Jul 1;124(7):962-9.

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.*

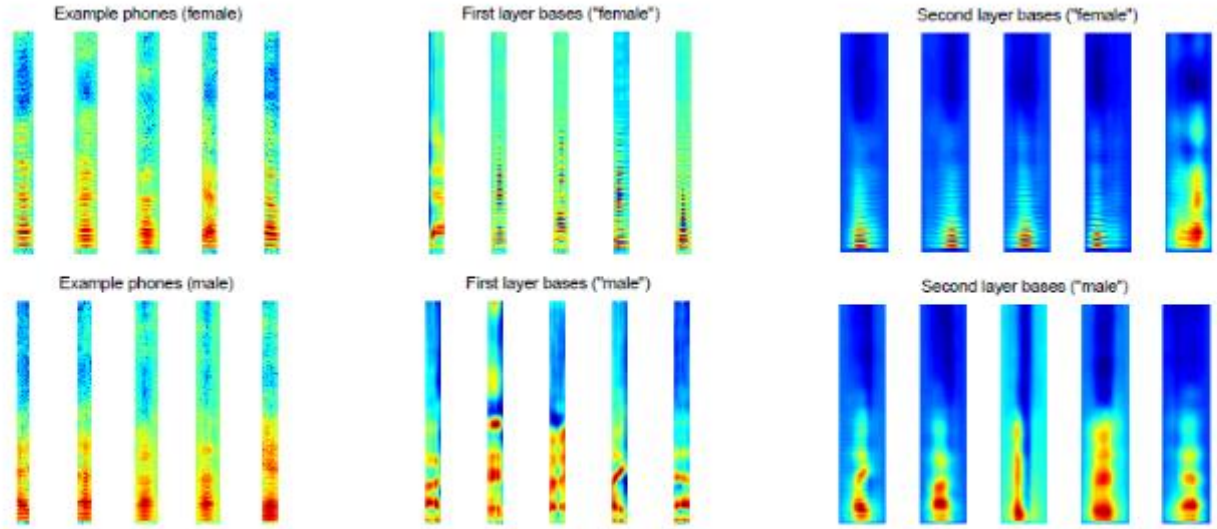


Figure 3: (Left) five spectrogram samples of "ae" phoneme from female (top)/male (bottom) speakers. (Middle) Visualization of the five first-layer bases that most differentially activate for female/male speakers. (Right) Visualization of the five second-layer bases that most differentially activate for female/male speakers.

A **spectrogram is heap map** 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.

Columns -- time interval of first-layer in the spectrogram space.
Rows -- frequency channels ordered from lowest to highest one.

Lee H, Pham P, Largman Y, Ng AY. Unsupervised feature learning for audio classification using convolutional deep belief networks. In Advances in neural information processing systems 2009, pp. 1096-1104.

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.

Lakhani P, Sundaram B. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology. 2017 Apr 24;284(2):574-82. https://pubs.rsna.org/doi/full/10.1148/radiol.2017162326

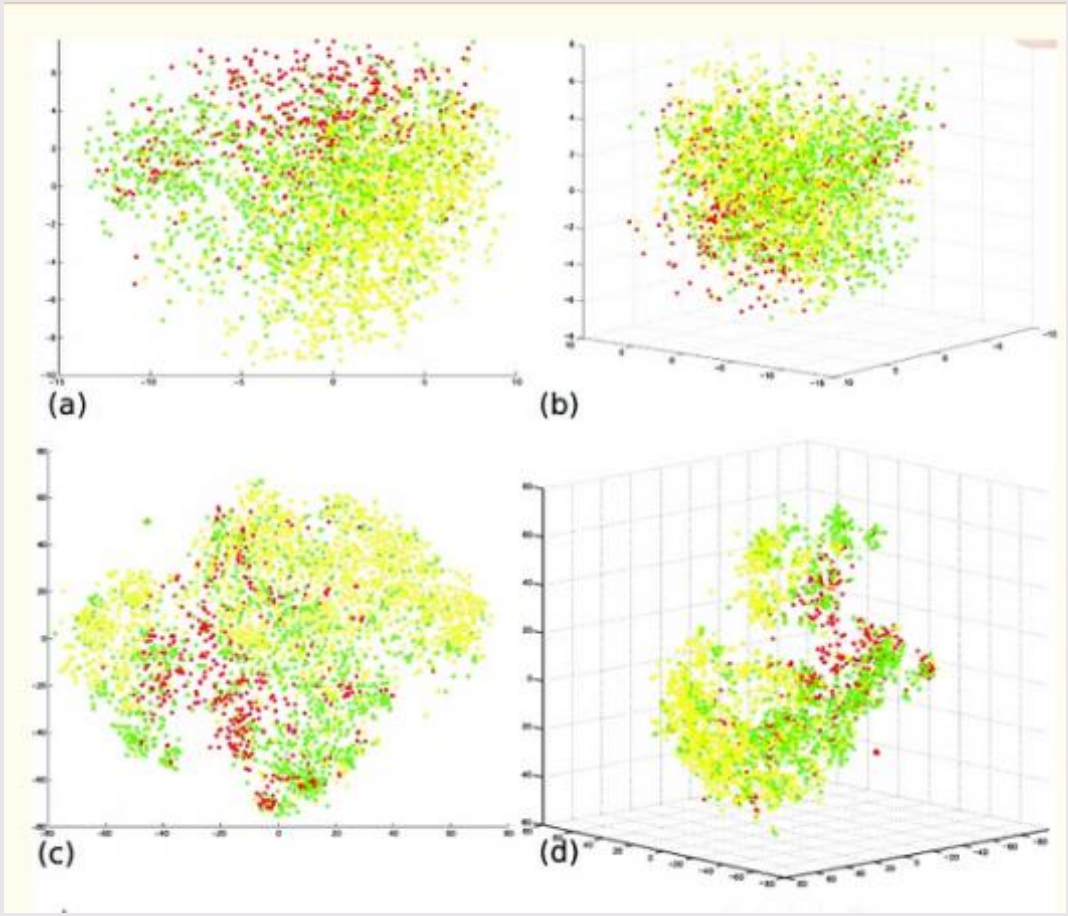
Nonlinear feature space dimension reduction in breast computer-aided diagnosis (CADx)

- **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.*

Jamieson, A.R.; Giger, M.L.; Drukker, K.; Lui, H.; Yuan, Y.; Bhooshan, N. Exploring Nonlinear Feature Space Dimension Reduction and Data Representation in Breast CADx with Laplacian Eigenmaps and t-SNE. Medical Physics. 37 (1): 339–351. 2010 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2807447/

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]*

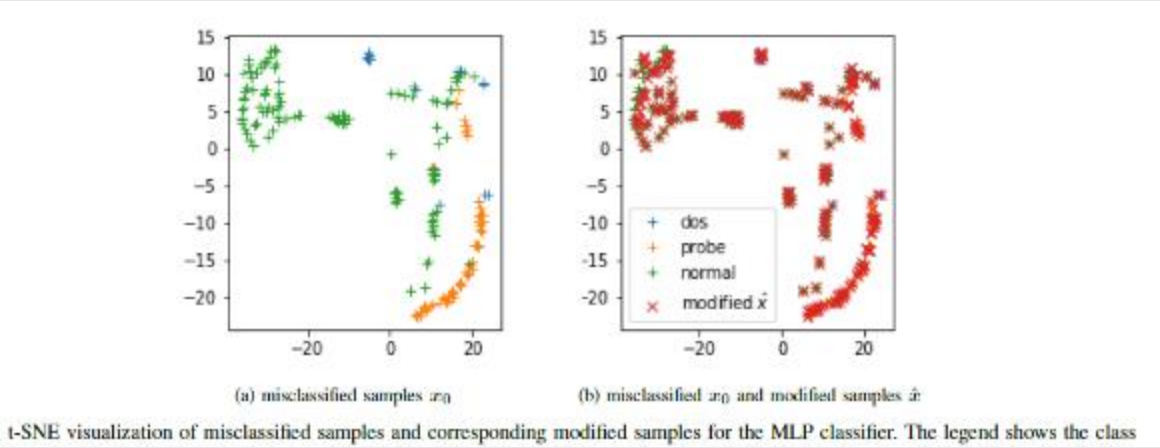


Jamieson, A.R.; Giger, M.L.; Drukker, K.; Lui, H.; Yuan, Y.; Bhooshan, N. Exploring Nonlinear Feature Space Dimension Reduction and Data Representation in Breast CADx with Laplacian Eigenmaps and t-SNE. Medical Physics. 37 (1): 339–351. 2010

van der Maaten, L.J.P.; Hinton, G.E. "Visualizing Data Using t-SNE". Journal of Machine Learning Research. 9: 2579–2605., 2008

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.*



t-SNE visualization of misclassified samples and corresponding modified samples for the MLP classifier. The legend shows the class

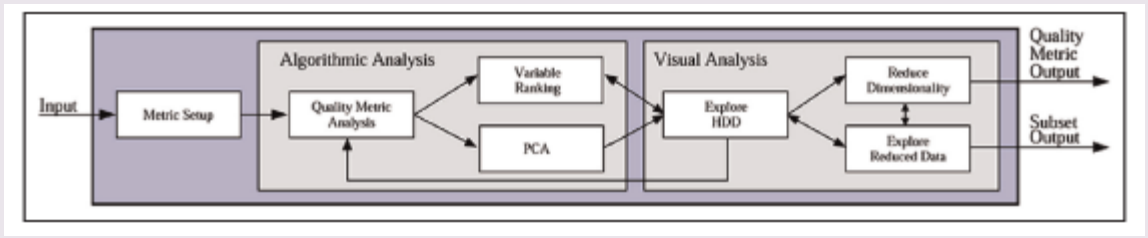
Marino DL, Wickramasinghe CS, Manic M. An adversarial approach for explainable AI in intrusion detection systems. In IECON 2018-44th Conference of the IEEE Industrial Electronics Society 2018 Oct 21 (pp. 3237-3243). IEEE.

Approaches to discover analytical ML models assisted by visual means

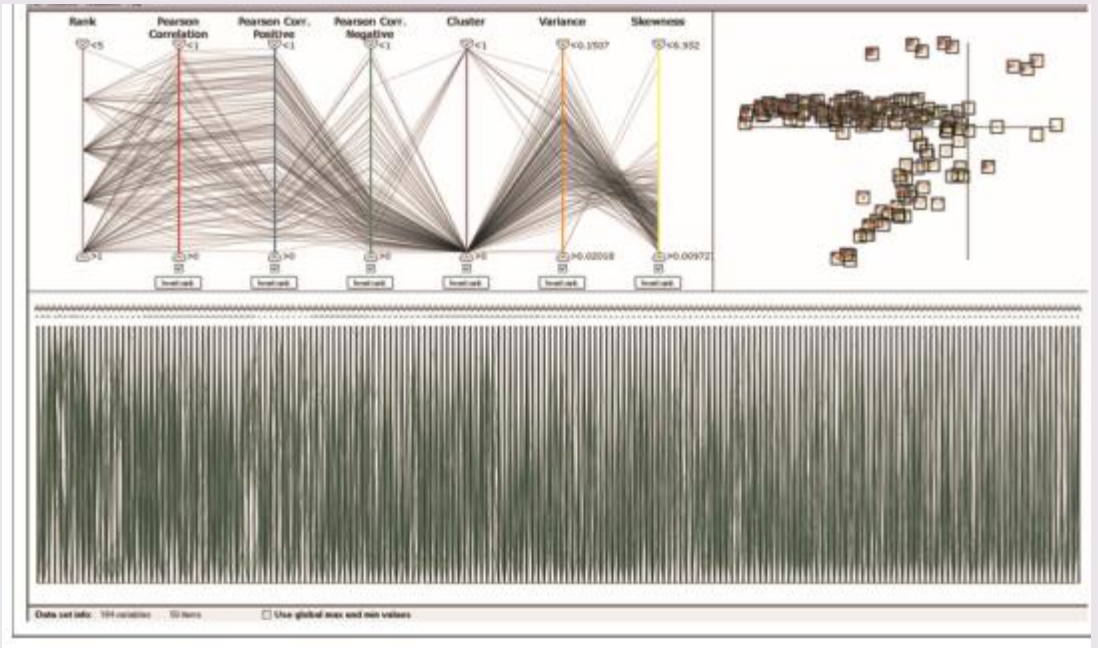
- Discovering visual ML models assisted by analytical ML algorithms, such as
  - *propositional and first order rules,*
  - *random forests,*
  - *CNN,*
  - *decision trees,*
  - *optimization based on genetic and other algorithms*

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.*



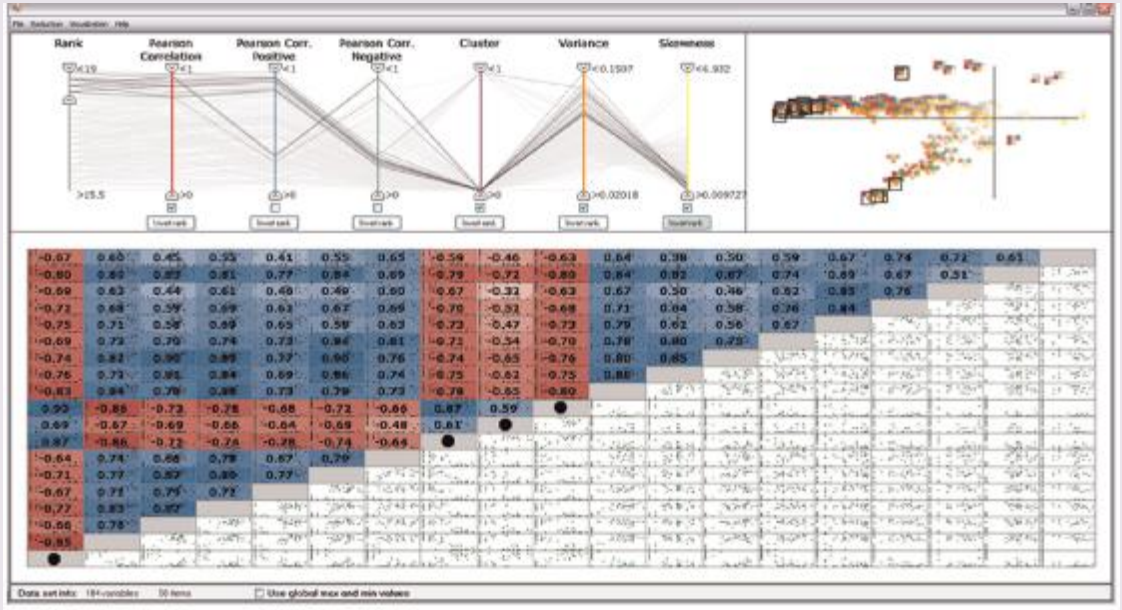
Workflow



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.

Fernstad SJ, Shaw J., Johansson J., Quality-based guidance for exploratory dimensionality reduction, Information Visualization 12(1) 44–64, 2013

Quality-based guidance for exploratory dimensionality reduction



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

Top – visualizations of quality metrics of variables using parallel coordinates and glyphs.

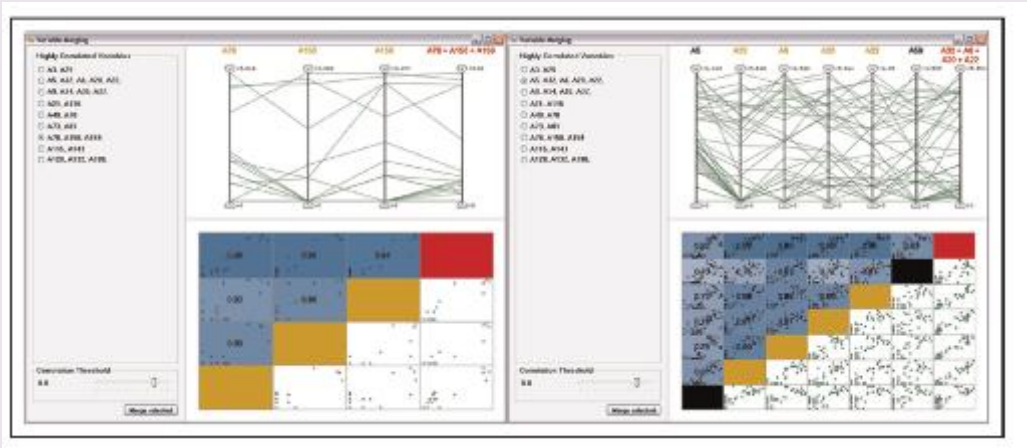
In **parallel coordinates**, axes are quality metrics, and polylines are the **variables**.

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.



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.*

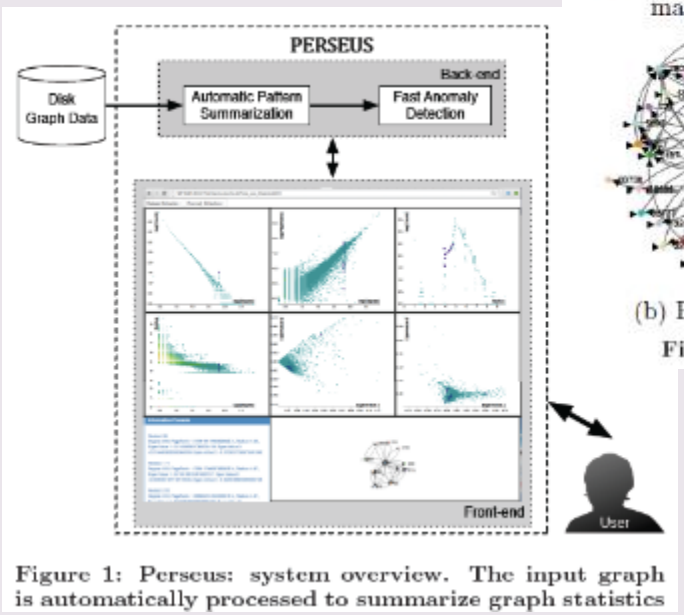
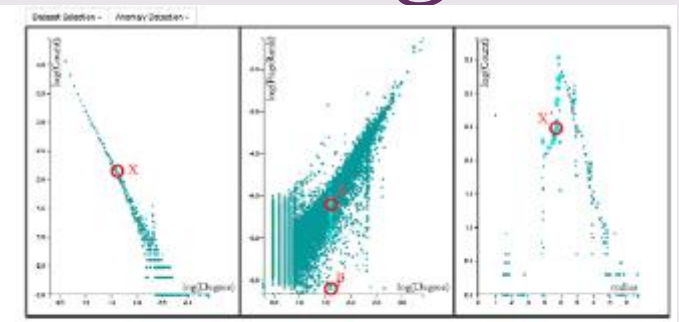
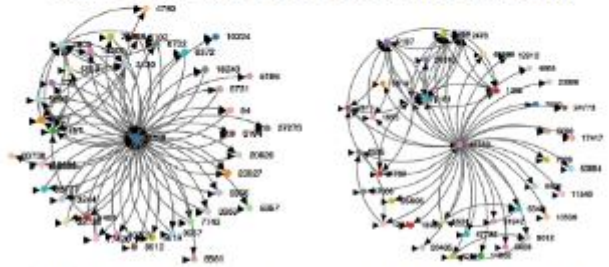


Figure 1: Perseus: system overview. The input graph is automatically processed to summarize graph statistics



(a) The selected point X in the degree distribution plot maps to the cyan points in the other two plots.



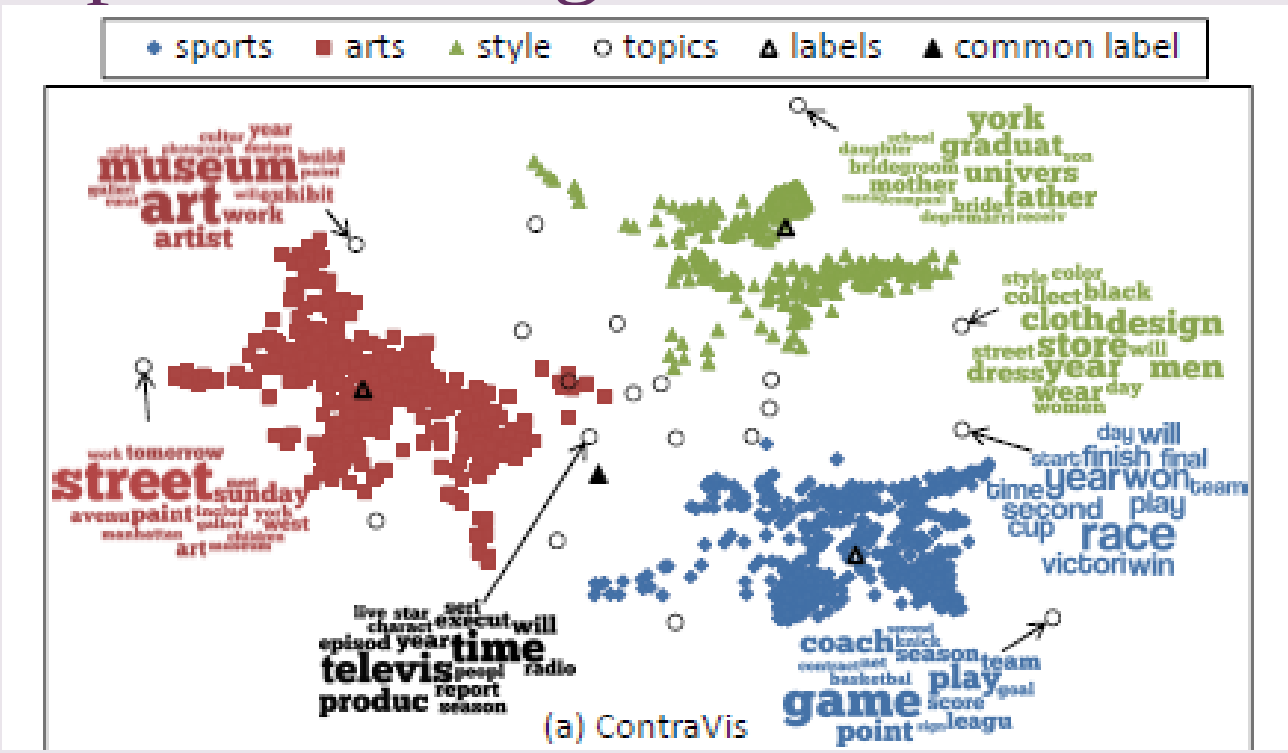
(b) Egonet of node A. (c) Egonet of node B.

Figure 3: User-guided anomaly detection.

D. Koutra, D. Jin, Y. Ning, C. Faloutsos. Perseus: An Interactive Large-Scale Graph Mining and Visualization Tool.*Proceedings of the VLDB Endowment (VLDB'15 Demo)*, August 2015. http://www.vldb.org/pvldb/vol8/p1924-koutra.pdf

Contrastive and Visual Topic Modeling for Comparing Documents

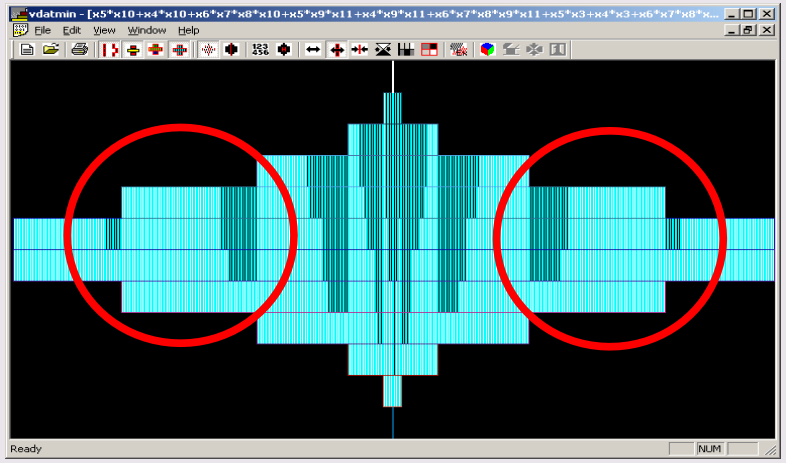
- Problem:
  - *How to express **similarities/differences** of 'labeled' document collections and visualize them efficiently?*
- Approach
  - *Learn hidden topics and **embeddings** for the documents, topics and labels for visualization.*
  - *Extract hidden **discriminative and common topics** across 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.*



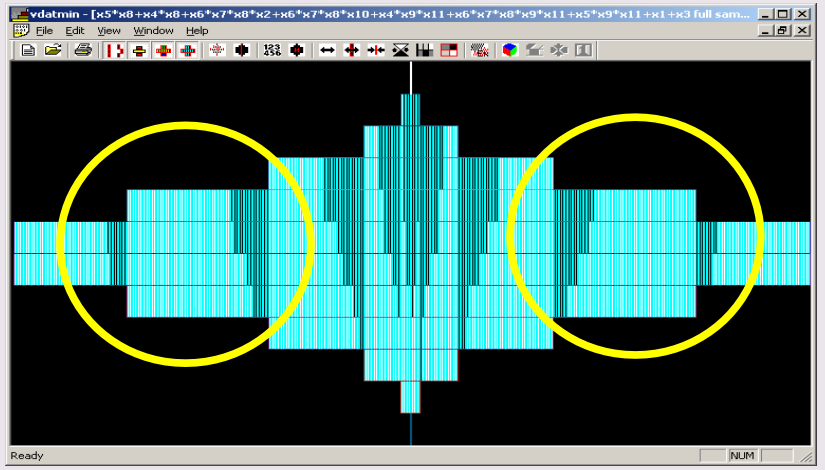
Contrastive analysis of sports, arts, style documents from NYTnews (20 topics).
ContraVis: (supervised joint method) finds latent discriminative topics and gives visual embedding of documents and topics.
Two discriminative topics per label shown in wordclouds (w/ respective colors) + one common topic (black);

Le T. & Akoglu L., ContraVis: Contrastive and Visual Topic Modeling for Comparing Document Collections, 2019, http://www.andrew.cmu.edu/user/lakoglu/pubs/19-www-contravis.pdf

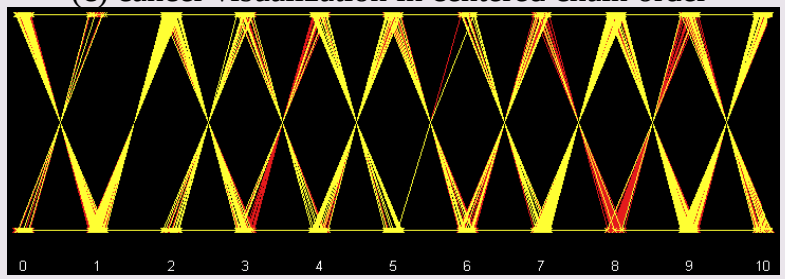
Monotone Boolean Function visualization vs. Parallel Coordinates for binary data



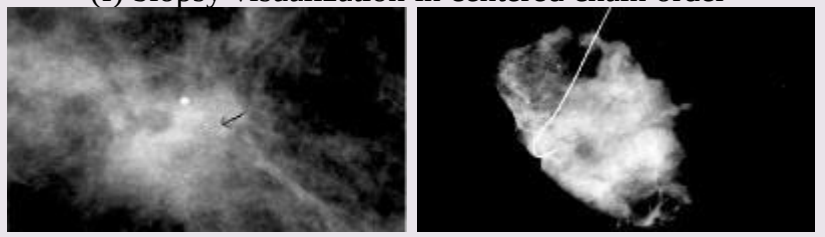
(c) cancer visualization in centered chain order



(f) biopsy visualization in centered chain order



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

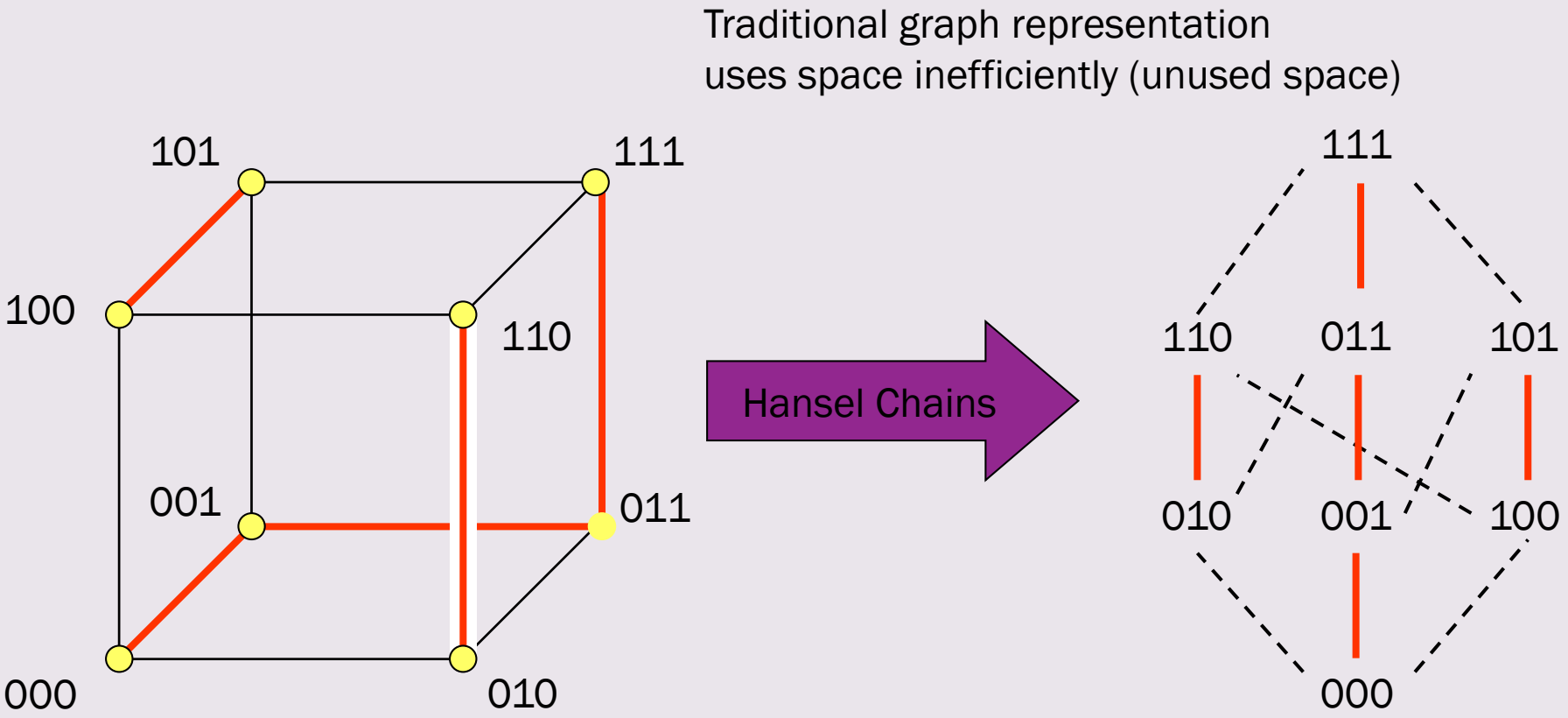


(k) Types of source X-ray mammography images used producing Boolean vectors

Kovalerchuk, B.; Delizy, F.; Riggs, L., E. Vityaev, Visual Data Mining and Discovery with Binarized Vectors, in: Data Mining: Foundations and Intelligent Paradigms (2012) 24: 135-156.

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



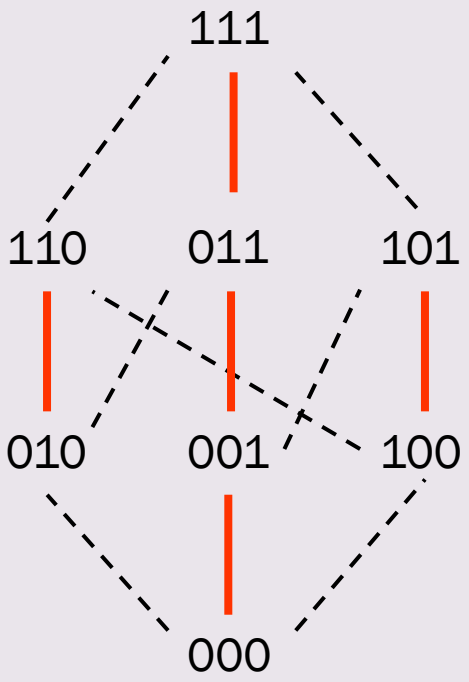
the concept of the monotone Boolean function from discrete mathematics [Korshunov, 2003, Keller, Pilpel, 2009]. Let $E^n=\{0,1\}^n$ be a binary *n*- dimensional cube then vector $\mathbf{y}=(y_1,y_2,\ldots,y_n)$ is no greater than vector $\mathbf{x}=(x_1,x_2,\ldots,x_n)$ from E^n if for every i $x_i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

$$\mathbf{x} \geq \mathbf{y} \Leftrightarrow \forall i\ x_i \geq y_i$$

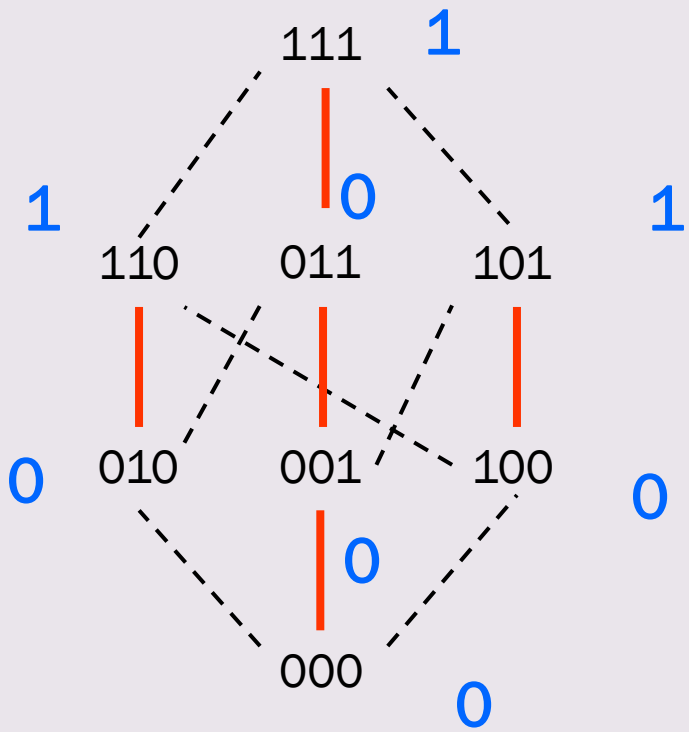
In other words, vectors **x** and **y** are ordered. In general relation ≥ for Boolean vectors in E^n is a *partial order* that makes E^n a *lattice* with a max element (1,1,…,1) and min element (0,0,…,0).
Boolean function f: $E^n \rightarrow E$ is called a *monotone Boolean function* if

$$\forall\ \mathbf{x} \geq \mathbf{y} \Rightarrow \mathbf{f(x)} \geq \mathbf{f(y)}.$$

Space structure



Data in the space structure



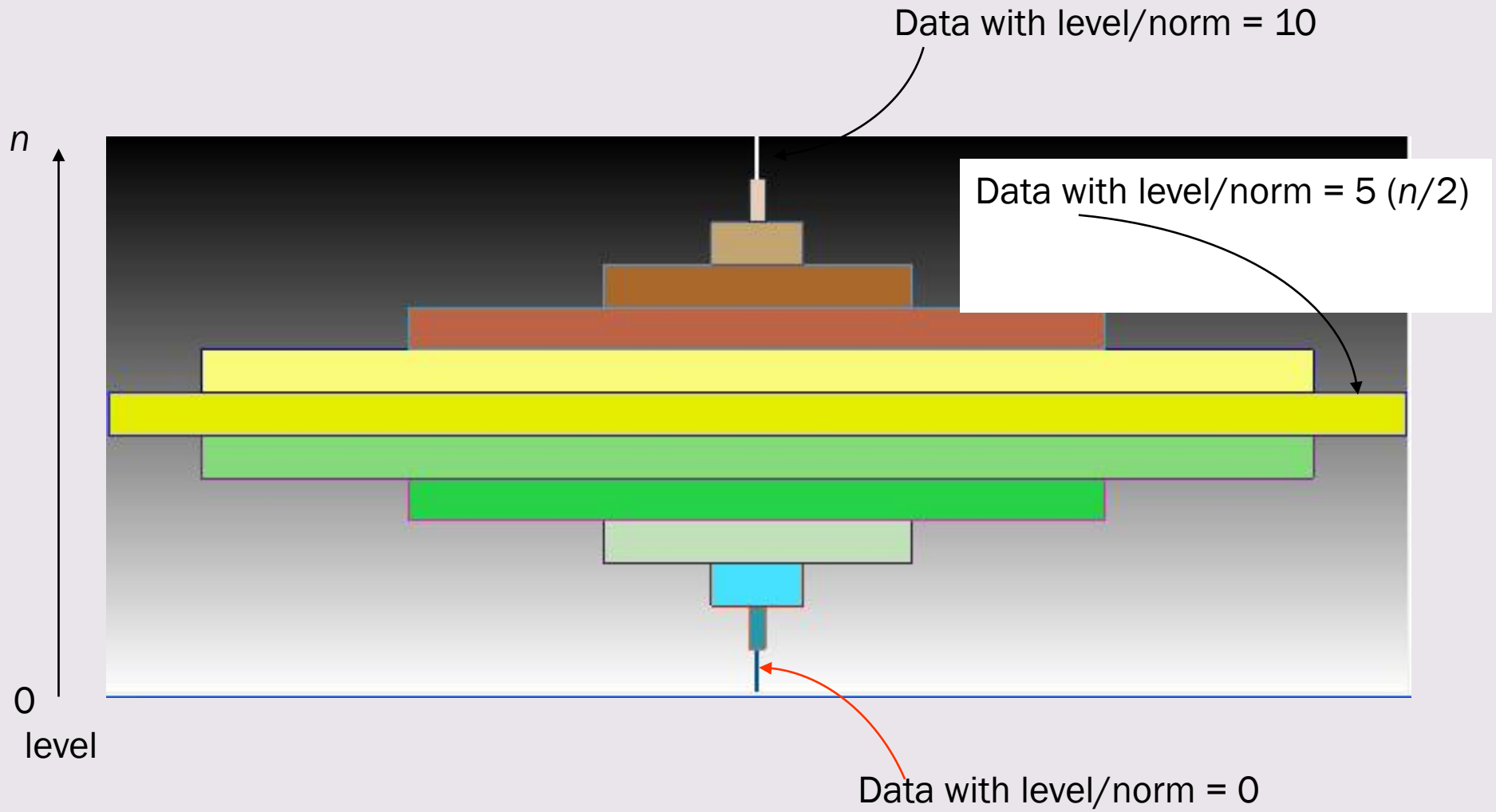
This monotonicity property implies two expansion properties for function f:

$$\mathbf{x} \geq \mathbf{y}\ \&\ f(\mathbf{y})=1 \Rightarrow \mathbf{f(x)}=1,\quad \mathbf{x} \geq \mathbf{y}\ \&\ f(\mathbf{x})=0 \Rightarrow \mathbf{f(y)}=0,$$

A Basic Example

P_1 process numeric based vector location

- MDF without data



48-D and 96-D data in CPC-Stars, Radial Stars and Paral Coordinates

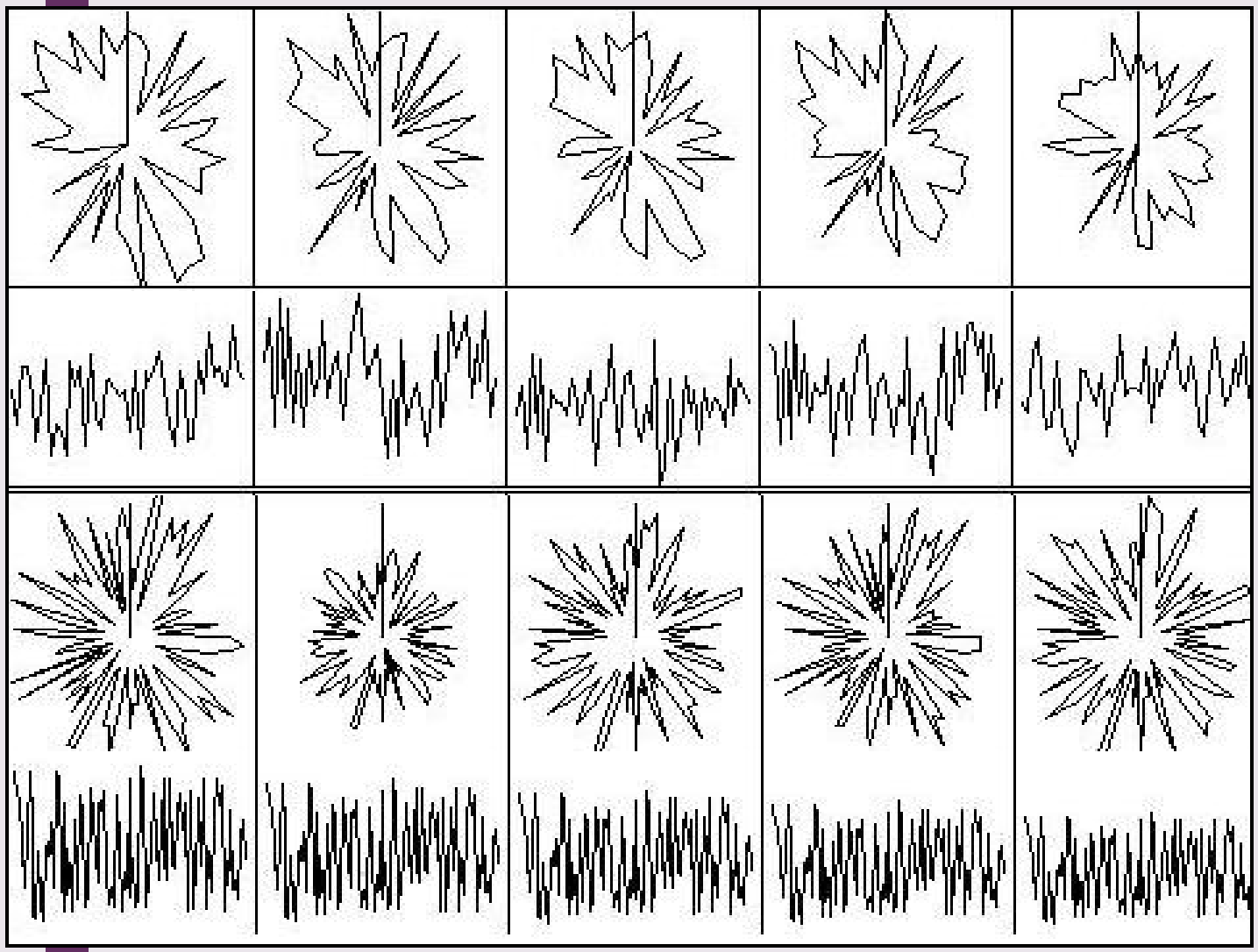


Figure 6.3. Examples of corresponding figures: stars (row 1) and PCs lines (row 2) for five 48-D points from two tubes with m = 5%. Row 3 and 4 are the same for dimension n=96.

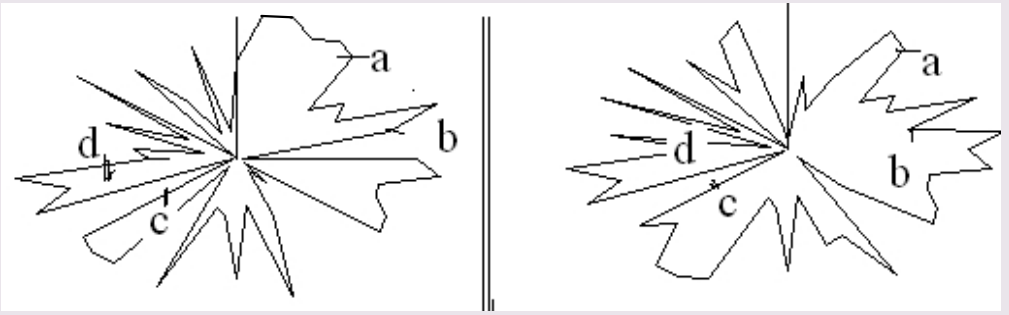


Figure 6.4. Two stars with identical shape fragments on intervals [a,b] and [d,c] of coordinates.

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

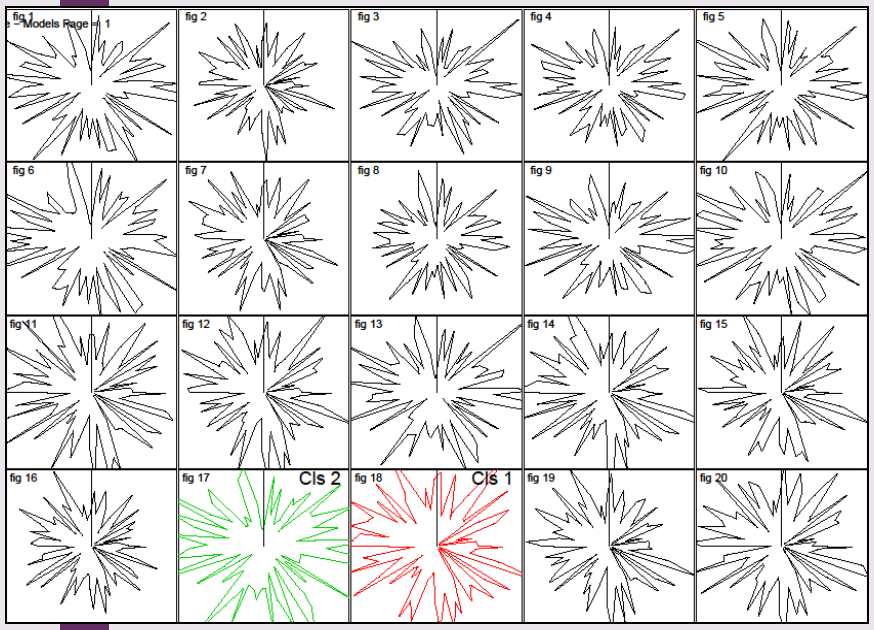
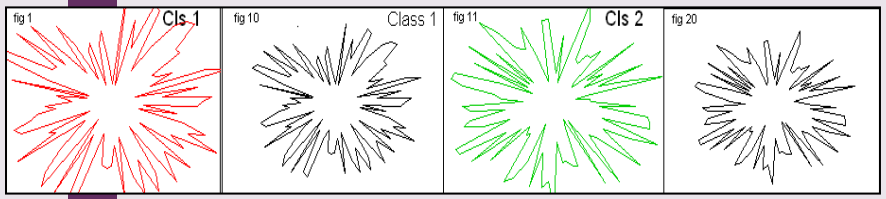
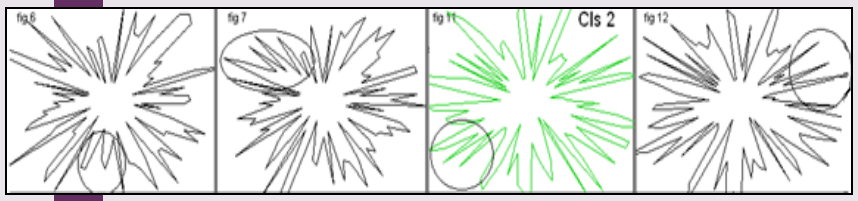


Figure 6.7. Twenty 160-D points of 2 classes represented in star CPC with noise 10% of max value of normalized coordinates (max=1) and with standard deviation 20% of each normalized coordinate.



(a) Initial 100-D points without noise for Class (Hyper-tube) #1 and Class (Hyper-tube) #2



(b) 100-D points with multiplicative noise: circled areas are the same as in upper star.

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

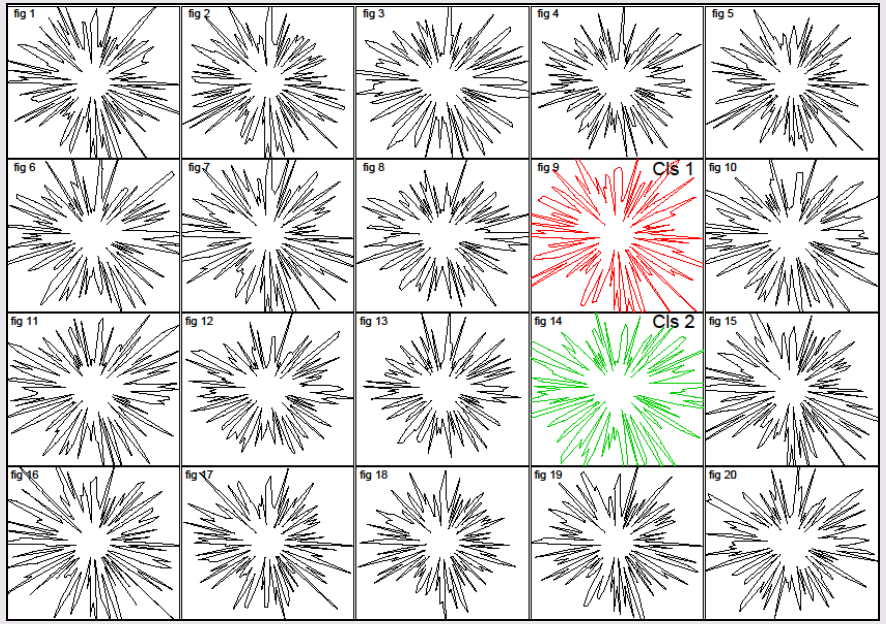


Figure 6.8. Twenty 160-D points of 2 classes represented in Radial Coordinates with noise 10% of max value of normalized coordinates (max=1) and with standard deviation 20% of each normalized coordinate.

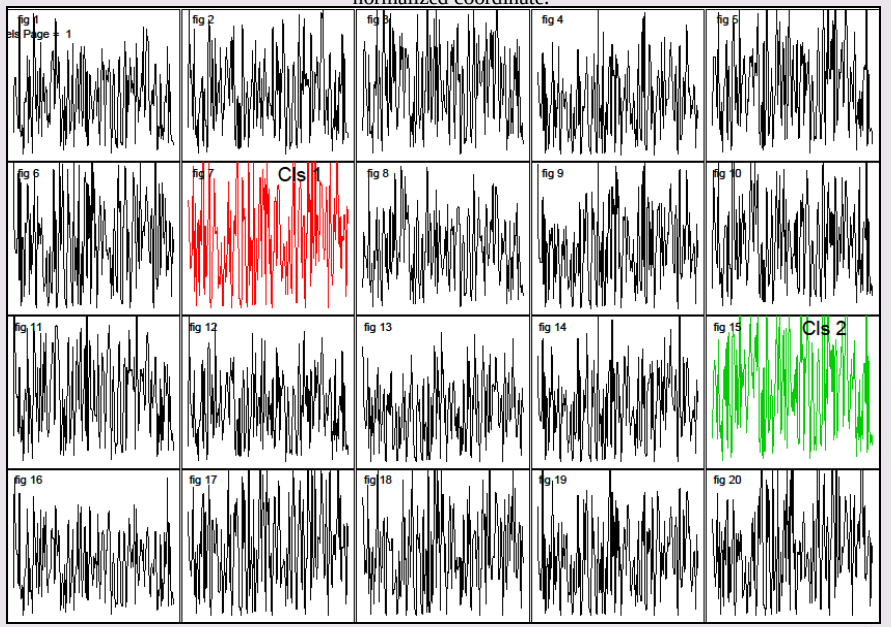


Figure 6.9. Twenty 160-D points of 2 classes represented in Parallel Coordinates with noise 10% of max value of normalized coordinates (max=1) and with standard deviation 20% of each normalized coordinate.

Human abilities to discover patterns in 170-D data

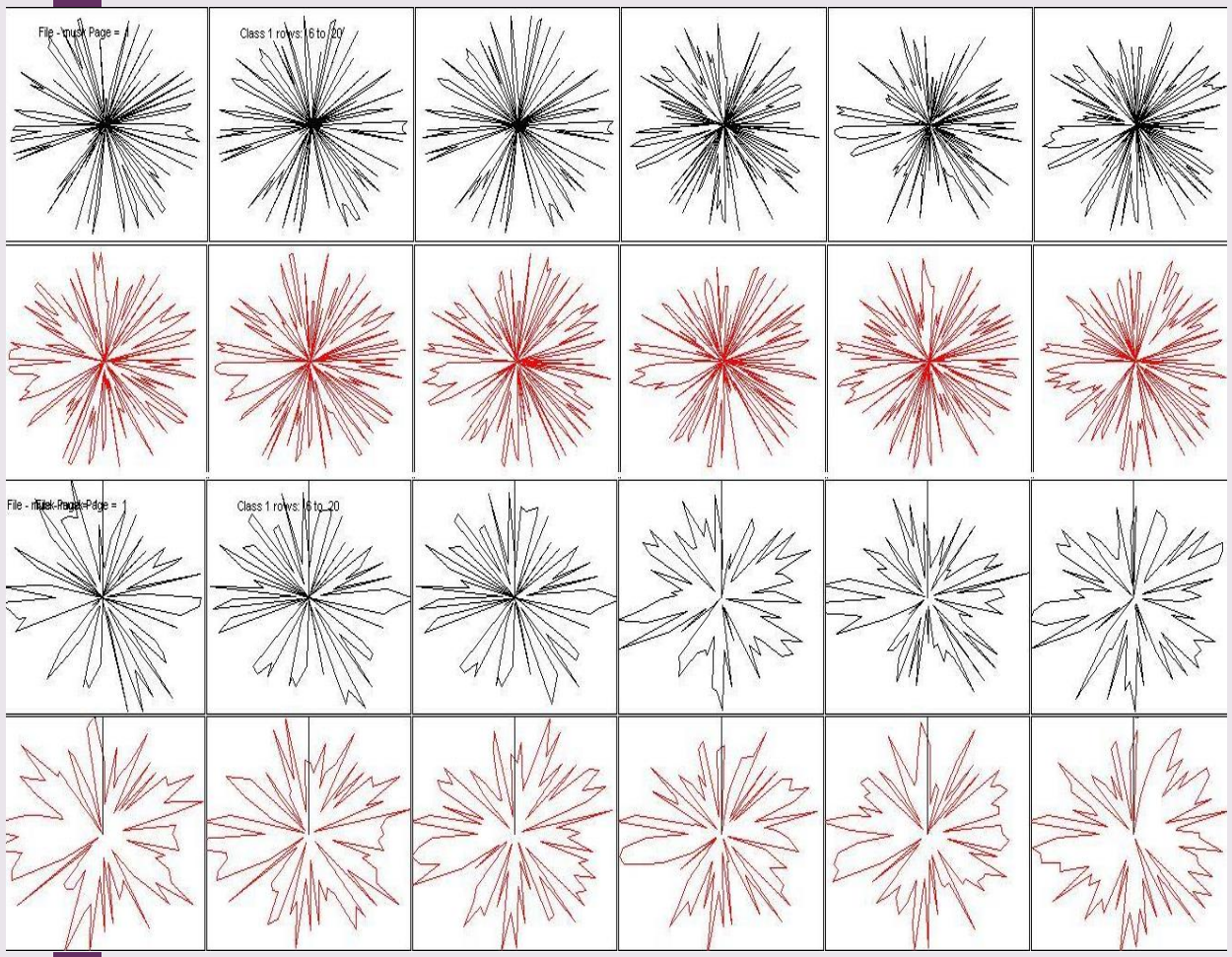


Figure 6.17. Traditional 170-D stars: class “musk” (first row) and class “non-musk chemicals” (second row). CPC 170-D stars from the same dataset: class “musk” (third row) and class “non-musk chemicals” (forth row).

- 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. L
- 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 addition 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

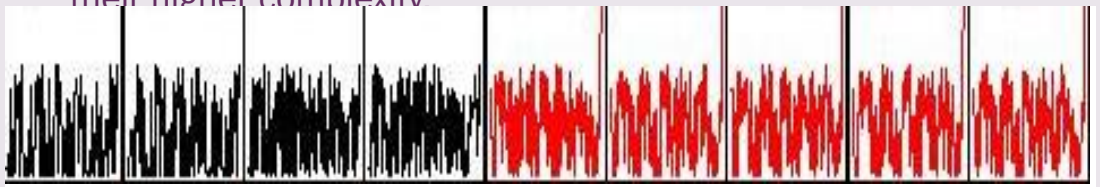


Figure 6.18. Nine 170-dimensional points of two classes in Parallel Coordinates

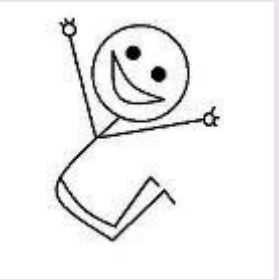
Conclusion

- 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
  - Kovalerchuk B. Visual Knowledge Discovery and Machine Learning, Springer 2018.
- 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]
  - Muggleton, S.H. (1991). Inductive logic programming. New Generation Computing. **8** (4): 295–318.
  - Lavrac, N.; Dzeroski, S. (1994). Inductive Logic Programming: Techniques and Applications. New York: Ellis Horwood.
  - Mitchell T. (1997) Machine Learning, McGraw-Hill
  - Kovalerchuk B., Vityaev E. (2000) Data Mining in Finance, Advances in Relational and Hybrid Methods, Springer, 2000.

REFERENCES

Aggarwal CC. Towards effective and interpretable data mining by visual interaction. ACM SIGKDD Explorations Newsletter. 2002 Jan 1;3(2):11-22.

Andrienko N, Lammarsch T, Andrienko G, Fuchs G, Keim D, Miksch S, Rind A. Viewing visual analytics as model building. In: Computer Graphics Forum 2018 Sep (Vol. 37, No. 6, pp. 275-299).

Angelini M, Aniello L, Lenti S, Santucci G, Ucci D. The goods, the bads and the uglies: Supporting decisions in malware detection through visual analytics. In: 2017 IEEE Symposium on Visualization for Cyber Security (VizSec) 2017 Oct 2 (pp. 1-8). IEEE.

Cashman D, Humayoun SR, et al., Visual Analytics for Automated Model Discovery. arXiv preprint arXiv:1809.10782. 2018

Endert A, Ribarsky W, Turkay C, Wong BW, Nabney I, Blanco ID, Rossi F. The state of the art in integrating machine learning into visual analytics. In: Computer Graphics Forum 2017 Dec (Vol. 36, No. 8, pp. 458-486).

Fuchs R, Waser J, Groller ME. Visual human+ machine learning. IEEE Transactions on Visualization and Computer Graphics. 2009 Nov;15(6):1327-34.

Jiang L, Liu S, Chen C. Recent research advances on interactive machine learning. Journal of Visualization. 2018 Nov 12:1-7.

Jin, L., Koutra. D., ECOviz: Comparative Visualization of Time-Evolving Network Summaries. ACM SIGKDD IDEA workshop 2017. https://www.lisajin.com/static/p40-jin.pdf

Johansson J. and Forsell C., Evaluation of parallel coordinates: Overview, categorization and guidelines for future research. IEEE Trans. on Visualization and Computer Graphics, 22(1):579–588, 2016.

Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS. Deep learning for visual understanding: A review. Neurocomputing. 2016 Apr 26;187:27-48.

Dimara E, Bezerianos A, Dragicevic P. Conceptual and methodological issues in evaluating multidimensional visualizations for decision support. IEEE Tr. on Vis. & Comp. Graphics. 2018 24(1):749-59.

Kehrer J, Hauser H. Visualization and visual analysis of multifaceted scientific data: A survey. IEEE transactions on visualization and computer graphics. 2013 Mar;19(3):495-513.

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., Visual Knowledge Discovery and Machine Learning, Springer, 2018.

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., Visual Knowledge Discovery and Machine Learning, Springer, 2018.

Kovalerchuk, B., Grishin, V. Reversible Data Visualization to Support Machine Learning. In: S. Yamamoto and H. Mori (Eds.): Human Interface and the Management of Information. Interaction, Visualization, and Analytics, LNCS 10904, Springer, pp. 45-59, 2018.

Kovalerchuk B., Gharawi A., Decreasing Occlusion and Increasing Explanation in Interactive Visual Knowledge Discovery, In: S. Yamamoto and H. Mori (Eds.) Human Interface and the Management of Information. Interaction, Visualization, and Analytics, LNCS 10904, Springer , pp. 505–526, 2018.

Kovalerchuk B., Neuhaus, N. Toward Efficient Automation of Interpretable Machine Learning. In: 2018 IEEE International Conference on Big Data, pp. 4933-4940, Seattle, Dec. 10-13, 2018 IEEE.

Kovalerchuk. B., Schwing J. (eds), Visual and Spatial Analysis: Advances in Visual Data Mining, Reasoning, and Problem Solving, Springer, 2005,

Koutra, D., Di Jin, Yuanchi Ning, Christos Faloutsos. Perseus: An Interactive Large-Scale Graph Mining and Visualization Tool.*Proceedings of the VLDB Endowment (VLDB'15 Demo)*, August 2015. Le T. & Akoglu L., ContraVis: Contrastive and Visual Topic Modeling for Comparing Document Collections, 2019,

Molnar, C. Interpretable Machine Learning, https://christophm.github.io/interpretable-ml-book/r-packages-used-for-examples.html

Montavon G, Samek W, MNguyen A, Yosinski J, Clune J. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. arXiv preprint arXiv:1602.03616. 2016 Feb 11

Muhammad T, Halim Z. Employing artificial neural networks for constructing metadata-based model to automatically select an appropriate data visualization technique. Applied Soft Computing. 2016 Dec 1;49:365-84.

.

üller KR. Methods for interpreting and understanding deep neural networks. Digital Signal Processing. 2018 Feb 1;73:1-5.

Poulet F. Towards effective visual data mining with cooperative approaches. In: Visual Data Mining 2008 (pp. 389-406). Springer, Berlin, Heidelberg.

Sacha D, Kraus M, Keim DA, Chen M. VIS4ML: An Ontology for Visual Analytics Assisted Machine Learning. IEEE transactions on visualization and computer graphics. 2019 Jan;25(1):385-95.

Schirrmeister RT, Springenberg JT, Fiederer LD, et al., Deep learning with convolutional neural networks for EEG decoding and visualization. Human brain mapping. 2017 Nov;38(11):5391-420.

Seifert C, Aamir A, Balagopalan A, Jain D, et al., Visualizations of deep neural networks in computer vision: A survey. In: Transparent Data Mining for Big and Small Data 2017 (pp. 123-144). Springer,.

Turkay C, Laramee R, Holzinger A. On the challenges and opportunities in visualization for machine learning and knowledge extraction: A research agenda. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction 2017 Aug 29 (pp. 191-198). Springer, Cham.

Yosinski J, Clune J, Nguyen A, Fuchs T, Lipson H. Understanding neural networks through deep visualization. In: the Deep Learning Workshop,31st International Conference on Machine Learning, Lille, France, 2015, arXiv preprint arXiv:1506.06579. 2015 Jun 22.

Wongsuphasawat K, Smilkov D, Wexler J, Wilson J, Mané D, Fritz D, Krishnan D, Viégas FB, Wattenberg M. Visualizing dataflow graphs of deep learning models in tensorflow. IEEE transactions on visualization and computer graphics. 2018 Jan;24(1):1-2.

Questions?

BORIS KOVALERCHUK

www.cwu.edu/~borisk

borisk@cwu.edu

More information