The final version was published by Springer:
S. Yamamoto and H. Mori (Eds.): HIMI 2018, LNCS 10904, pp. 45–59, 2018.
https://doi.org/10.1007/978-3-319-92043-6_4

Reversible Data Visualization to Support Machine Learning

Boris Kovalerchuk[1], Vladimir Grishin[2]

[1] Central Washington University, Department of Computer Science,
Ellensburg, WA, 98922, USA
borisk@cwu.edu
[2] ViewTrend Int., 1001 Colonial Ave SE
Palm Bay, FL 32909, USA

Abstract. An important challenge for Machine Learning (ML) methods such as the Support Vector Machine (SVM), and others, is the selection of the structure of ML models for given data. This paper shows that the abilities of the pure analytical ML methods to address this challenge are limited. It is due to the fundamental nature of the ML methods, which rely on the available training data, which can result in overgeneralized or overfitted model. In the proposed visual analytics approach, domain experts are put into the "driving seat" of the ML model development to control the model overgeneralization and overfitting. In this approach, domain experts work interactively with multidimensional data, and the ML data classification models, presented in the lossless reversible visualizations. This paper shows that it enhances the ML classification models, and decreases the use of external and irrelevant-to-the-domain assumptions in the ML models.

Keywords: Multidimensional data, visualization, machine learning, classification, reversible lossless visualization.

1 Introduction

An important challenge for Machine Learning (ML) methods, such as the Support Vector Machine (SVM) and others, is the *selection and evaluation of the structure of machine learning models*. It includes (1) a class of models to be explored (linear, nonlinear with a type of non-linearity), and (2) *characteristics* of the models such as the set of possible kernels. Usually these structures are chosen empirically, based on problem knowledge and multiple runs of the algorithm with *limited data visualization*. Typically, many categories of characteristics of the ML models are external for the given task and data. They are imposed by the ML method not derived from the data. This process can lead to *inadequate models*. Such models can lack interpretation, provide wrong predictions on new unseen data, can be overfitted or overgeneralized.

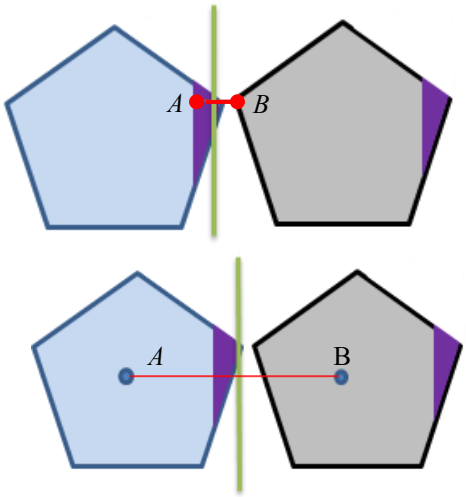
In the current machine learning practice, visualization is commonly used, for illustration and explanation of the ideas, of many algorithms such as the Support Vector Machine, or the Fisher Linear Discriminant Analysis (LDA), but much less for the actual discovery of the n-D rules, models due to the difficulties to adequately represent the n-D data in 2-D.

Besides, some ML methods use only a part of the available information to construct the models. For instance, an ML algorithm can use only the cases from each class, which are close to the cases of the opposing class, in the training data. Such very limited usage of training dataset information can prohibit getting models that are more efficient. Visualization can help to discover situations where it can decrease the efficiency of ML models. It is illustrated in Fig.1.

Fig. 1 shows an example where SVM uses only two closest support vectors *A* and *B*, while LDA uses all training cases to construct a line that discriminates classes. In Fig. 1a we use the geometric interpretation of the linear SVM [Bennett, Campbell, 2000; Bennett, Bredensteiner, 2000] that shows that SVM uses the closest support vectors of the two classes.

Respectively, in Fig. 1a linear SVM uses a red line that connects *two closest support vectors* (SV) *A* and *B* from opposing classes (blue and grey areas that constitute training data *D*). This red line is a basis of the green discrimination line, which bisects the red line in the middle and is orthogonal to it.

In contrast, in Fig. 1b, the simplified Fisher LDA uses the *average points of all of the training data* of each class (points *A* and *B*), connects them with the red line. Then the orthogonal green line bisects it in the middle. The green line serves as the discrimination line. In Fig. 1, two algorithms produce different green discrimination lines, which are *error free on the training data*. However, the LDA, which used the training data more fully to build the discrimination line, has *less error* on violet *validation data* of the left class. While visualization in Fig. 1 clearly and quickly shows this, discovering it analytically would require extensive work to build both models completely, and generate quite specific validation data, which will allow detecting a lower accuracy of the SVM model.



(a) Linear SVM classifier based on two closest support vectors *A* and *B*.

(b) LDA linear classifier based on centers *A* and *B* of classes.

Fig. 1. Two classes classified by linear SVM and simplified LDA [Kovalerchuk, 2018].

This paper proposes a way to improve the construction of the ML models, which use the more complete information in the visual form, visualized in 2-D to get a more accurate classification. This approach is based on the reversible lossless General Line Coordinates (GLC) [Kovalerchuk, Grishin, 2017; Kovalerchuk, 2018]. It opens the

opportunity for (1) visual discovery of linear separatability of the classes, (2) ensuring the finding of it by the classifiers, and (3) getting additional visual confidence in the linear separation by the domain users in the understandable form.

Next, the actual boundary between the classes can be *non-linear, even when the linear separation exists*, because the linear separation can *severely overgeneralize* the training data as we show below. The proposed visual analytics approach supports the estimation of the level of nonlinearity of boundaries, and respectively selecting better parameters of the non-linear ML models.

Consider this overgeneralization challenge, of the ML models, using the 4-D Iris data [Lechman, 2013]. These data consist of 150 cases of three classes of Iris: Setosa, Versicolor and Virginica. Each case is represented by sepal and petal length and width as a 4-D point. Fig. 2 shows the examples of Iris petals of these classes. Later on, in one of our experiments, petals are modeled by ovals of respective length and width.

Setosa Versicolor Virginica

Fig.2. Examples of Setosa, Versicolor and Virginica Iris petals.

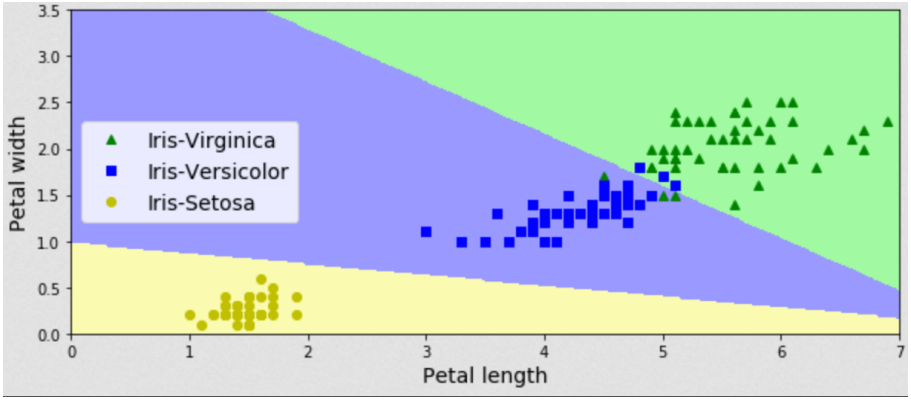
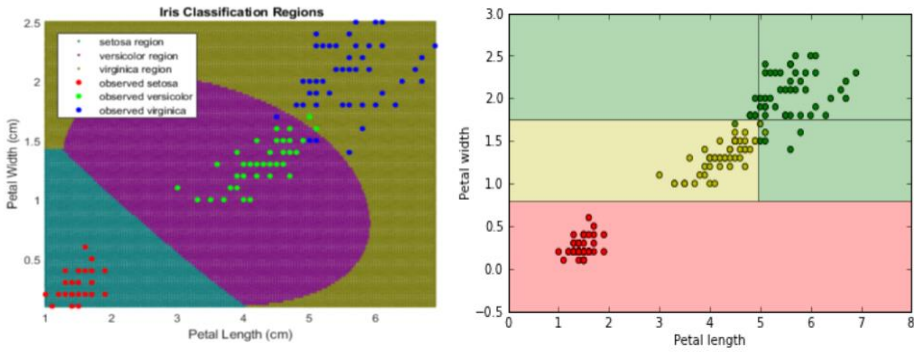


Fig.3. Example of logistic regression classification of these classes of Iris [Big data, 2018].

Figs. 3 and 4 show the examples of *logistic regression* [Big data, 2018], *SVM* [FitSVM, 2018] and *Decision Tree* [Taylor, 2011] classification of these classes of Iris, provided in these references as machine learning tutorials and lectures. In these figures, the results are visualized in two dimensions (petal length and width). While all of three classification models are quite accurate, on the given cases, they *very differently generalize and classify the data outside the given cases*.

Moreover, all of them *significantly overgeneralize* given 150 cases relative to human generalization that we report in the experiment section later. In fact, according these models, irises with all possible combinations of petal length and width exist. For

instance, the logistic regression and the Decision Tree allow Setosa petal to be more than 4 times longer than those in the Setosa training data, and Versicolar petal to be more than 4 times shorter than in the Versicolar the training data (see Figs. 3 and 4).



(a) SVM [Matlab, FITCSVM, 2018] (b) Decision Tree [Taylor, 2011]

Fig.4. Example of SVM and Decision Tree classification of three classes of Iris.

2 Approach

Historically in ML, overgeneralization was associated with the situation, where only the positive examples (example of one class) are available, which "do not provide information for preventing overgeneralization of the inferred concept." [Carbonell, Michalski, Mitchell, 1984]. It is contrasted with the situation when both positive and negative examples are available, where "negative examples *prevent overgeneralization* (the induced concept should never be so general as to include any of the negative examples)" [Carbonell, Michalski, Mitchell, 1984]. However, as we see in Figs. 3 and 4 the presence of the cases of the opposing classes prevent only such an *extreme overgeneralization* as the direct overlap of classes.

A computational approach to control overgeneralization is proposed in [Pham, Triantaphyllou, 2008] based on partitioning the space of the training data according to the data density. The expansion of the training data area is made proportional to the density of points in the area measured by *homogeneity degree* HD defined in that paper. The HD approach assumes that areas that have higher HD can be expanded more up to the point where expanded areas of opposing classes reach each other. The application of this approach to Iris data will definitely make more conservative generalization than shown in Figs. 3 and 4. However, it still can be overgeneralization as we discuss below.

The major drawbacks of this approach are: (1) the density of training data in each area is low in a high dimensional space; (2) the higher density in the area does not justify larger expansion of this area.

This heuristic rule is not derived from the given task, but it is an *external hypothesis*. It is also not a type of question that can be answered by the domain expert easily to clarify the relation of the larger expansion hypothesis to the domain task.

Our approach is different. It is putting the domain expert into the "*driving seat*" of the ML model development. The premise is that interaction of the domain experts

with the visualization of the n-D data and the alternative discrimination functions, models in these visualizations will enable the making of better ML classification models, and decrease in the use of external and irrelevant-to-the-domain assumptions in the ML models.

The feasibility of this approach is partially supported by our previous studies where the domain expert (radiologist) was able to identify deficiencies of the rules discovered by ML algorithms, when these rules were presented in the understandable and visual forms [Kovalerchuk et al, 2000, 2012].

Selecting and evaluating the class of ML models requires identifying:

- the *type of the ML model* to be discovered – linear or non-linear,
- the *form of non-linearity*, and
- the level of generalization *confidence* of the discovered ML model.

The model can be overfitted, overgeneralized, or a right one from the viewpoint of the domain expert. Here we explore the abilities of the visual approach, combined with analytical means, to test and increase the confidence in the ML model including finding the areas of high confidence.

Assume that we discovered a linear model *M*, which separates the given training and validation data with 100% accuracy. Can we assign this model *M* the highest confidence, and apply it to predict the class of *new unseen data* with high confidence? How to check that we can use the model *M*, with high confidence for such new data? Another situation happens when a linear model *M* separates the training and validation data with, say, 60% accuracy. How to modify *M* to be able to apply it to predict the class of new unseen data with high confidence?

The uncertainty in both situations is coming from the fact that typically with high-dimensional data we do not know for sure *how representative* given training and validation data, that are used to build model *M*, for prediction on the new unseen data. In this situation, making an assumption on the probability distribution of high-dimensional data outside of the training data rarely can be of high confidence. As a result, we cannot assign a reliable probabilistic level of confidence to the predictions outside of the training data.

Moreover, there are situations where some unseen data do not belong to any of the training data class and their classification *must be refused* by a classifier. As we have seen in Figs. 3 and 4 it is not done for Iris data by all three ML algorithms.

Another example is classification of letters. Let letters *A* and *B* represented as 16-D data [Lichman, 2013] be classified successfully by some ML model that does not refuse to classify any case. Thus, any letter, encoded as a 16-D point, will be classified as *A* or *B* which is a vast overgeneralization.

Thus, the fundamental task is developing methods to test and increase confidence in the ML models. While the most efficient approach for this is *adding new training data and attributes*, however, it is not possible, in many real world situations.

An alternative idea is adding other types of *additional information*. The examples are below. The expert can bring the information that data must be within hyperspheres, centered about the given templates, or the expert may select and analyze the n-D points close to the border between the classes, and/or far away from the training data, and tell that those points must belong to other classes than the model *M* suggests. In the experiment section we explore this alternative idea.

The generation of new data can be done analytically without visualization or interactively with visualization. The analytical way can be quite challenging to implement. Let we want to generate n-D points that are close to the classification line. This task is *ill-posed mathematically* without setting constrains around that line.

Next, we need to deal with the infinite number of n-D points in the area limited by a threshold that also needs to be set up. Therefore, we need to limit the number of points that will be randomly generated. Those assumptions are difficult to set up and justify formally and rigorously. In contrast, the human expert can see and select such points visually if an appropriate visualization of the n-D data and n-D classifier are provided. This will support the finding of the anomaly and the generation of the confidence/non-confidence evaluation of the ML model faster.

3 Experiments

This section presents experiments conducted to check the proposed approach, to see how the domain experts can limit the overgeneralization of ML models, using visualization. It is conducted as a series of experiments with the participants (computer science students).

Each participant worked with the visualization plots, assigning the level of human confidence between 0 and 10, for classification of the different cases and models. The data used in these experiments were selected in a way to ensure that students can serve as reasonable "experts" in classifying these data.

Fig. 5 shows data of the two classes, where two disjoint areas represent class 1. In experiment 1 (Fig. 5, left), participants evaluate their confidence in classification of the red point to class 1 or class 2 using the confidence scale from 0 to 10 with 10 indicating the max of confidence.

In experiment 2 (Fig. 5, right), participants evaluate their confidence in classification of the blue point to class 1 or 2 using the same confidence scale. These experiments can clarify the level of human confidence in classification of highly contested points.

Fig. 5. Setting of experiments 1(left) and 2 (right)

Fig. 6 shows the setting of experiment 3, where participants evaluate their confidence in alternative separation lines between the two classes using the same confidence scale from 0 to 10. The first alternative is a linear discrimination line with a small margin and the second one is a non-linear discrimination line. This experiment can clarify the human preference between simpler, but less accurate line on the left vs. the more complex, and likely more accurate discrimination line on the right.

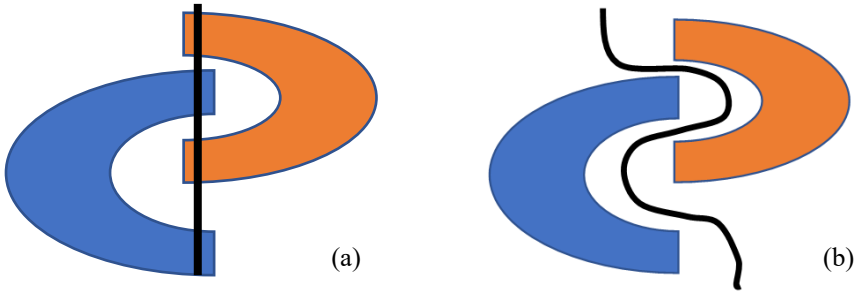


Fig. 6. Setting of experiment 3 to evaluate the two alternative separation lines between the classes.

Table 1 presents the quantitative result of these experiments, with 7 participants (Computer Science students). In the experiments 1 and 2, participants are highly uncertain in classifying the red and blue points into either class. The confidence is just about 50:50.

Table 1. Results of experiments 1-3.

	Experiment 1		Experiment 2		Experiment 3		
Class	mean	Stdev	mean	stdev	Alternative	mean	stdev
Class 1	4.14	1.81	5.14	1.81	(a)	5.00	2.20
Class 2	6.86	1.25	6.29	1.48	(b)	7.14	2.17

The experiment 3 shows that participants prefer a more accurate non-linear discrimination line (b), while the automatic ML algorithms may stop by finding the linear discrimination line in the alternative (a), because many ML algorithms, in the attempt to balance simplicity and accuracy, assume that simpler lines are more robust. While this general assumption is valid, it is not task specific, and for the given data, it may or may not be applicable.

We conducted five more experiments, with 11 participants (computer science students). In these experiments, we used the same Iris data from the UCL Machine Learning repository [Lichman, 2013].

In experiments 4-6, the participants evaluated their confidence in the petal classification of the two petals (denoted as A and B) into class 1 or 2. An oval defined by its length and width represents each petal. In the experiments 5-8, participants evaluated petals represented by 2-D points (x,y) where x is the length and y is the width of the petal, in the 2-D Cartesian coordinates. Thus, in these experiments participants observe only the parameters of the petals (not the actual petals simulated as ovals in experiment 4).

The participants marked their confidence in the same confidence scale from 0 to 10. The petals A and B have been selected in the middle between the two classes in the contested "grey" area where the intuitive classification is difficult. The additional information in experiment 4 is given as ovals, which represent the centers of classes and the two closest petals from opposing classes (see Fig. 7).

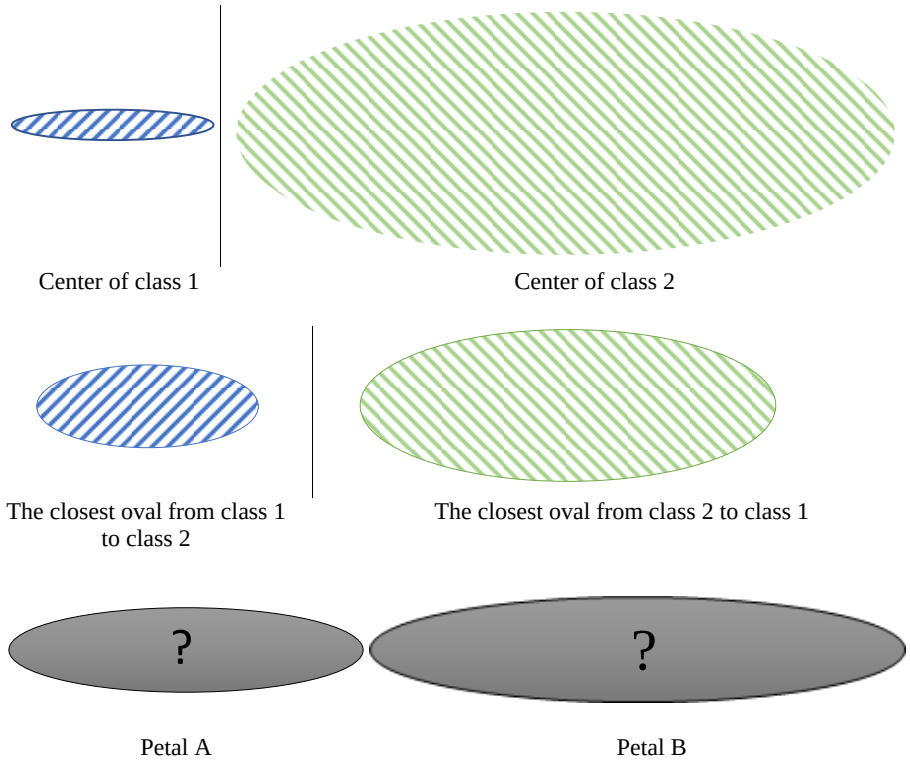


Fig. 7. Setting of experiment 4.

The settings of the experiments 5-7 are shown in Fig. 8. Participants evaluated the points A and B in experiments 5 and 6, and points C and D in experiment 7. The red and black lines are shown to the participants only in experiment 6.

The quantitative results of experiments 4-8 are shows in Tables 2-4. The analysis of Tables 2-4 allows making the following conclusions. In experiment 4, participants are more confident that both ovals *A* and *B* are in class 2. However, the difference between the means of the confidences for opposing classes are very small: 5.27 vs. 4.91 for oval *A* and 5.45 vs. 4.93 for oval *B* with quite large standard deviations from 1.91 to 2.9 (see Table 2). In other words, in average the confidence is about 50:50 in experiment 4.

In experiment 5, participants are more confident in classifying point (oval) *A* into class 1 and point (oval) to *B* into class 2: 7.82 vs. 2.27 for *A* and 6.36 vs. 3.45 for point (oval) *B* standard with standard deviations from 1.4 to 2.45.

Thus, this result is quite different from the result of experiment 4. In experiment 5, point *A* is classified into the different class (class 1) with high confidence (7.82 vs. 4.91 for class 1 in experiment 4). In experiment 5, point *B* is classified into the same class as in experiment 4, but with slightly higher confidence (6.36 vs. 5.45).

The important difference between these experiments is that in experiment 4 participants observe *actual ovals*, but in experiment 5 they observe only 2-D points that represent *parameters* of ovals (length and width) in 2-D Cartesian Coordinates. The other difference is that in experiment 4 participants observe only 6 ovals, but in

experiment 5 they observe ten times more (about 60) 2-D points that represent 102 ovals. With all these differences, the conclusion is that the use of *parameters* of actual objects gives *more confidence* than the use of *actual objects* in classifying difficult cases from the "grey" border area between classes.

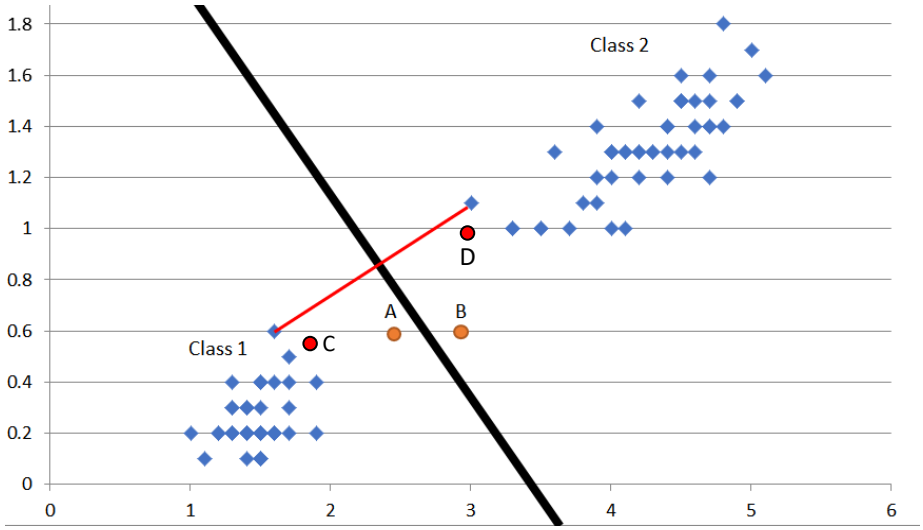


Fig. 8. 3. Setting of experiments 5-7.

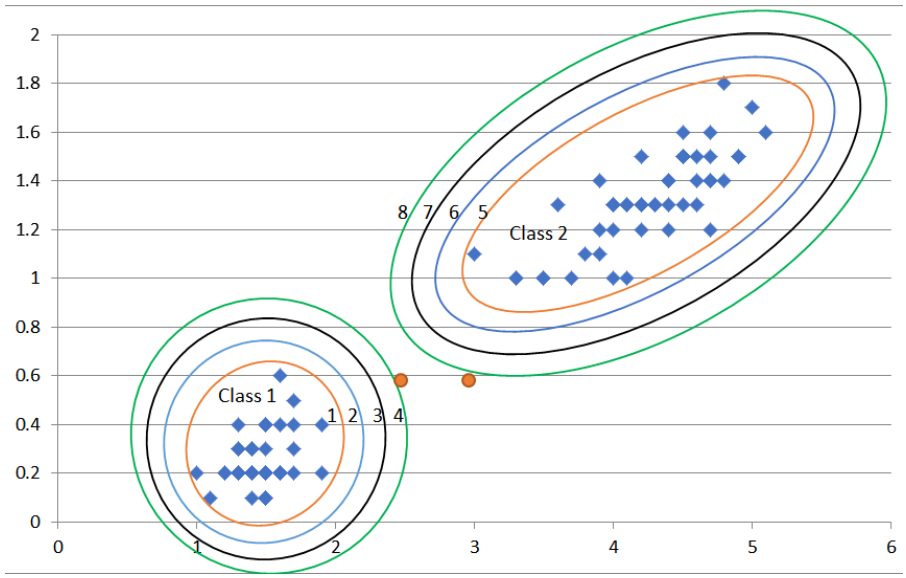


Fig. 9. Setting of experiment 8.

Table 2. Results of Experiments 4-6

		Experiment 4		Experiment 5		Experiment 6	
		mean	stdev	mean	stdev	mean	Stdev
Class 1	Oval A	4.91	2.31	7.82	1.40	7.82	2.95
Class 1	Oval B	4.73	2.70	3.45	2.19		
Class 2	Oval A	5.27	1.91	2.27	1.48		
Class 2	Oval B	5.45	2.90	6.36	2.46	5.82	3.59

Table 3. Results of Experiment 7

		mean	stdev
Class 1	Oval C	9.64	0.64
Class 2	Oval D	9.27	1.71

Table 4. Results of Experiment 8.

		mean	stdev
Class 1	Oval 1	9.73	0.45
Class 1	Oval 2	8.91	0.79
Class 1	Oval 3	6.91	2.81
Class 1	Oval 4	5.91	2.91
Class 2	Oval 5	9.78	0.42
Class 2	Oval 6	9.00	0.82
Class 2	Oval 7	7.11	2.69
Class 2	Oval 8	6.22	2.7

Our hypothesis is that this result is not accidental, but it can be explained. Quite typically, experts have more work experience with actual objects than with selected parameters of those objects in some mathematical form. More experience with actual objects can lead to more justified judgements on classification of objects by observing the actual objects. In actual objects, experts can observe selected parameters differently, e.g., holistically.

Next, actual objects are a source of *additional parameters* that experts can use for classification of objects. Pure intuitively it makes more sense to tell 50:50 (i.e., refuse to classify objects in the "grey" area) than to classify those objects with a high confidence. This reasoning suggests that relying on human judgement in the setting of experiment 4 is *more reliable* than in the setting of experiment 5. In the future experiments it will be interesting to test this conclusion on other objects.

In comparison with experiment 5, in experiment 6, adding the black classification line does not change the relatively large confidence for point *A* (78.2%, stdev 2.95) to be in class 1, but slightly decreases the low confidence in point *B* to be in class 2, to 5.82 (with standard deviation 3.59) from 6.36, in experiment 5. Thus, presence of the tip (in the form of the classification line) does not change much the classification result and confidence. Moreover, the decrease in confidence for point *B* shows that the closeness of point *B* to the classification line alerts the participants on the risk the classification of *B* to class 2.

Experiment 7 consistently shows high confidence that point *C* is in class 1 (9.64%, std. 0.64); and point *D* is class 2 (9.27, std. 1.71). This means that the participants do not limit classes 1 and 2 by their convex hulls, because both points are outside of the respective convex hulls.

The design of experiment 8 is shown in Fig. 9, where participants assigned their confidence values to ovals 1-4 to class 1 and ovals 4-8 to class 2. Experiment 8 consistently shows decreasing confidence from inner ovals to outer ovals for both classes: from high confidence 9.73 with stdev 0.45 (class 1) and 9,78 with stdev 0.42

(class 2) to low confidence 5.91 with stdev 2.91 (class 1) and 6.22 with stdev 6.22 (class 2) of the outer ovals. For the outer ovals these numbers are close to the results of experiment 4 (5.27 for oval *A* and 5.45 for oval *B*) and less close to experiment 5 (7.82 for point *A* and 6.36 for point *B* and even further from experiment 6 for point *A* (7.82, stdev 2.95), but closer for point *B* (5.82, stdev 3.59).

The experiments presented in this section show that participants can control overgeneralization of data in a variety of visualization settings. All cases that are far away from training data were classified with low confidence, in essence, refused to be classified. This is in a sharp contrast with automatic ML models shown in Figs. 3 and 4, where no case was refused.

4. Reversible data visualization method to support Machine Learning

While the experiments in the previous section were done on 2-D data, in this section, we show the use of the lossless visualization of 4-D data to support finding a Machine Learning classification model.

The *lossless visualization approach for multidimensional data* that we use below is based on the concept of the *General Line Coordinates* (GLC) and one of its specific forms denoted as *Parametrized Shifted Paired Coordinates* (PSPC) [Kovalerchuk, 2018; Kovalerchuk, Grishin, 2017]. GLCs allow visualizing n-D data with full *preservation* of n-D information in 2-D. In this sense, GLC is *lossless* and *reversible* allowing restoring all n-D information from 2-D visualization of an n-D point presented in 2-D as a graph.

Fig. 10 shows Iris data of classes 1 and 2 using all 4 attributes x_1-x_4 represented in PSPC in 2-D. In PSPC each 4-D point $\mathbf{x}=(x_1, x_2, x_3, x_4)$ is represented as a line (arrow) from point (x_1, x_2) in Cartesian coordinates (X_1,X_2) to point (x_3, x_4) in Cartesian coordinates (X_3,X_4). In Fig. 10a, coordinates (X_3,X_4) are shifted relative to (X_1,X_2) in such way that the center of class 1 became a single 2-D point. Similarly, in Fig. 10b this shift is done in such way that the center of class 2 became a single 2-D point. See [Kovalerchuk, Grishin, 2017; Kovalerchuk, 2018] for details of this method.

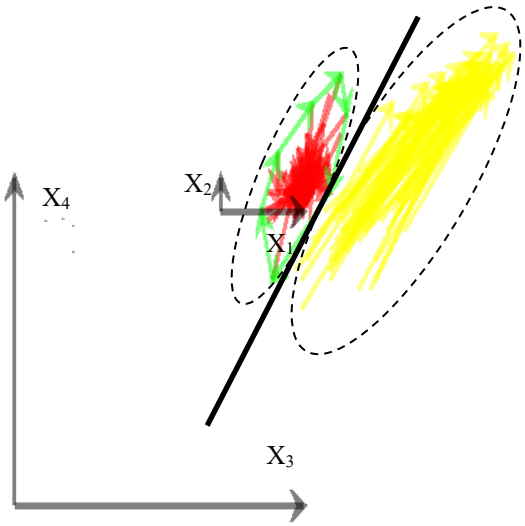
While Fig. 10 clearly shows that classes 1 and 2 are linearly separated by the black line, it would be an extreme overgeneralization to claim that every point, above that black line, is in class 1, and every point, below it, is in class 2. However, as we discussed in section 1, the popular ML algorithms such as the SVM, LDA and Decision Trees do such overgeneralization.

The GLC visualization of the n-D data allows the analysis, observing it in 2-D and setting control of the overgeneralization, using *implicit domain knowledge*. The first layer of generalization could be the convex hull, around the respective graphs of n-D points of classes (simple arrows in PSPC for 4-D data). For class one, the convex hull is shown in Fig 10a in green.

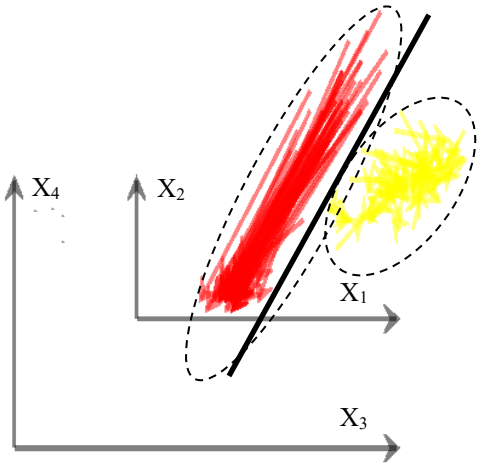
This figure shows the next layer as two dotted ovals. The GLC visualization system can produce it automatically or an analyst can do this interactively using implicit domain knowledge how far the unseen cases can differ from presented in the training data at the different levels of confidence.

Generalization beyond these dotted ovals, to larger ovals, will lead to overlap classes, as the visualization clearly shows. Thus, the analyst needs either to stop the

generalization at these ovals, or change the shape of the border (the model of the border).



(a) Iris data in PSPC anchored in class 1. The class 1 convex hull is in green.



(b) Iris data in PSPC anchored in class 2.

Fig. 10. 4-D data reversible visualization to identify ML model parameters.

The change of the model would mean expanding ovals only in the areas, where they do not touch each other. It is difficult to justify such change of the model. We simply do not have any information, to justify this change of the model, beyond our desire to avoid overlap, in the area where the ovals already touch each other.

For instance, ovals in Fig. 10b do not allow the Iris parameters x_i to be equal to zero. In contrast, the decision tree in Fig. 4b overgeneralizes, to such unrealistic Iris

cases, as well as to the cases with the petal length 4 times greater than in the training data.

The representation of n-D data as the 2-D *GLC graphs* such as shown in Fig. 10 has some similarity with the *manifold* approach. Commonly, the manifold approach uses a graph to find a surface in a subspace of much smaller dimension than the original n-D space [Gorban et al, 2007; McQueen et al., 2016]. This graph captures the distances between the nearest given n-D points to construct the surface. While this is valuable information, it is only partial information about the relations between the n-D points. It can be insufficient in discovering the patterns in the n-D data. Thus, this method is lossy; it preserves only a part of all the n-D information in a lower-dimension subspace. The manifolds are defined on n-D points not on 2-D graphs.

In contrast, the GLC graphs preserve all the information about the n-D points. While this is a difference between GLC graphs and manifolds, the commonalities between them are in shrinking the dimensions of the multidimensional data. The manifold can be expanded on the manifold surface, but not outside of this surface. This is like the expansion of the GLC graphs within the convex hulls, as shown in Fig. 10. Thus, manifolds control the overgeneralization, by allowing the data of the classes to be only on the surface of the manifold.

A similar control for the GLC graphs can be derived by analyzing the properties of these graphs within the oval. For instance, in Fig.10a arrows of the yellow class have a dominant direction and in Fig. 10b, the arrows of the red class have a dominant direction. This means that they occupy a fraction of the respective n-D areas.

This fraction of the area can be described mathematically similarly, to the manifold, e.g., by constructing a function of the two new attributes, angle and length of the arrow. In Fig. 10, the center of the oval that contains the GLC-graphs represents some n-D point. The n-D points, which are located around that n-D point, have their graphs within the oval. Thus, similar GLC graphs visualize similar n-D points. For hyper-cubes, it is proved mathematically in [Kovalerchuk, 2018].

The advantage of the GLC graph approach is that the domain experts can be in the 'driving seat" in analyzing and constructing the ML models controlling the generalization. In contrast, for the manifolds it is not clear how the domain experts can be in the same "driving seat" in constructing and limiting the manifolds. Moreover, the manifold can be higher dimension than 3, i.e., without natural visualization.

The restricted generalization in Fig. 10 illustrates the capabilities of the GLC visualization approach:

(1) Building interactively a *more accurate border between the classes* to avoid overgeneralization, and

(2) Getting a *better-explained description of the classes* that avoids the confusing description of the classes.

5 Conclusion

The proposed vision of the domain expert in the "driving seat" of model development depends on the success of the two important steps: (1) visualization of the data of the classes, making the classes separable in the visualization, and (2) abilities of the domain expert to generalize the data in these visualizations for

constructing and correcting the models. This paper shows the feasibility of both steps. More examples of the success of step (1) are presented in [Kovalerchuk, Grishin, 2017; Kovalerchuk, 2018; Kovalerchuk, Gharawi, 2018]. Can we ensure that it will always be successful? The answer is the same no, as for the analytical ML models. For some data, successful ML models do not exist. If accurate enough analytical ML exists for the given data, then the chances that the visualization model will exist will also be higher.

In general, visualizations of the n-D points, as GLC graphs in 2-D, allow the observation of all of their values in all the subspaces, including the overlap in each subspace. This supports the discovery of the efficient ML models including the support vectors in the SVM and the subspaces, where the data classes are separable.

Discovering visually just one such subspace is sufficient for solving the ML problem for the given data. Such effective tools include permuting and reversing the coordinates, in combination with the analytical search, for the efficient classification rules, and the visualization of them in GLC.

Holistic shapes of graphs [Grishin, Soula, 2003] allow their comparison to be more effective in the selection of subspaces. Besides that, the graphs give plenty of information, about the relationship between the parameters, in the subspaces and between the subspaces. These graphs are represented, in the different GLCs for the n-D data visualization, giving additional information for the mutual properties of data classes, relative to the linear and non-linear separation.

The future studies are expanding the class of ML models, which can benefit from the GLC visualization of the multidimensional data, and the ML models.

References

1. Bennett KP, Campbell C. Support vector machines: hype or hallelujah? ACM SIGKDD Explorations Newsletter. 2000;2(2):1-13.
2. Bennett KP, Bredensteiner EJ. Duality and geometry in SVM classifiers. In ICML 2000 Jun 29: 57-64.
3. Big Data and Machine Learning, 2018, http://www.cnblogs.com/luweiseu/p/7826679.html
4. FITCSVM, Mathworks, 2018, https://www.mathworks.com/help/stats/fitcsvm.html?s_tid=gn_loc_drop
5. Carbonell JG, Michalski RS, Mitchell TM. An overview of machine learning. In: Machine Learning, Volume I 1983 (pp. 3-23).
6. Gorban AN., Kégl B.**,** Wunsch, DC., Zinovyev A.(Eds.), Principal Manifolds for Data Visualisation and Dimension Reduction, Lecture Notes in Computer Science and Engineering (LNCSE), Vol. 58, Springer, Berlin – Heidelberg – New York, 2007. ISBN 978-3-540-73749-0
7. Grishin, V., Soula A., Pictorial Analysis: A Multi-resolution Data Visualization for Monitoring and Diagnosis of Complex Systems. 2003. International Journal of Information Science. Vol. 152, pp. 1-24.
8. Kovalerchuk B., Vityaev E., Ruiz J. Consistent knowledge discovery in medical diagnosis, IEEE Engineering in Medicine and Biology, v. 19, n. 4, pp. 26-37, 2000.
9. Kovalerchuk B, Delizy F, Riggs L, Vityaev E. Visual data mining and discovery with binarized vectors. In: Data Mining: Foundations and Intelligent Paradigms 2012 (pp. 135-156), Springer Berlin Heidelberg.

10. Kovalerchuk, B., Grishin, V. Adjustable general line coordinates for visual knowledge discovery in n-D data. Inf. Vis. 2017, doi:10.1177/1473871617715860.
11. Kovalerchuk B. Visual Knowledge Discovery and Machine Learning, 2018, Springer Nature, Cham, Switzerland.
12. Kovalerchuk, B. Gharawi, A., Decreasing occlusion in interactive visual knowledge discovery, In Human-Computer Interaction Intern. In: S. Yamamoto and H. Mori (Eds.): HIMI 2018, LNCS 10904, pp. 505–526, 2018, Springer.
13. Lichman, M. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013
14. McQueen J., Meila, M., VanderPlas J., Zhang Z., Mmegaman: Manifold Learning with Millions of points, 2016, https://arxiv.org/abs/1603.02763v1
15. Taylor, J., Stat 2002, Data Mining, Stanford, 2011, http://statweb.stanford.edu/~jtaylo/courses/stats202/trees.html
16. Pham HN, Triantaphyllou E. The impact of overfitting and overgeneralization on the classification accuracy in data mining. In: Soft computing for knowledge discovery and data mining 2008 (pp. 391-431). Springer, Boston, MA.