



HHS Public Access

Author manuscript

Proc IEEE Int Conf Big Data. Author manuscript; available in PMC 2019 December 01.

Published in final edited form as:

Proc IEEE Int Conf Big Data. 2018 December ; 2018: 3513–3521. doi:10.1109/BigData.2018.8621952.

Interactive Machine Learning by Visualization: A Small Data Solution

Huang Li¹, Shiaofen Fang¹, Snehasis Mukhopadhyay¹, Andrew J. Saykin², and Li Shen³

¹: Department of Computer & Information Science, Indiana University Purdue University Indianapolis

²: Indiana University School of Medicine

³: University of Pennsylvania Perelman School of Medicine

Abstract

Machine learning algorithms and traditional data mining process usually require a large volume of data to train the algorithm-specific models, with little or no user feedback during the model building process. Such a “big data” based automatic learning strategy is sometimes unrealistic for applications where data collection or processing is very expensive or difficult, such as in clinical trials. Furthermore, expert knowledge can be very valuable in the model building process in some fields such as biomedical sciences. In this paper, we propose a new visual analytics approach to interactive machine learning and visual data mining. In this approach, multi-dimensional data visualization techniques are employed to facilitate user interactions with the machine learning and mining process. This allows dynamic user feedback in different forms, such as data selection, data labeling, and data correction, to enhance the efficiency of model building. In particular, this approach can significantly reduce the amount of data required for training an accurate model, and therefore can be highly impactful for applications where large amount of data is hard to obtain. The proposed approach is tested on two application problems: the handwriting recognition (classification) problem and the human cognitive score prediction (regression) problem. Both experiments show that visualization supported interactive machine learning and data mining can achieve the same accuracy as an automatic process can with much smaller training data sets.

Keywords

interactive machine learning; visual data mining; multi-dimensional data visualization; user interaction

I. Introduction

Visualization, particularly multi-dimensional data visualization, has been playing an increasing important role in data mining and data analytics. This transformation of visualization from data viewing to being an integrated part of the analysis process led to the birth of the field of visual analytics [1]. In visual analytics, carefully designed visualization processes can effectively “decode” the insight of the data through visual transformations and interactive exploration. Many successful applications of visual analytics have been published in recent years, ranging from bioinformatics and medicine to engineering and social science.

These success examples demonstrate that visualization is a powerful tool in data analytics that needs to be seriously considered in any big data application. On the other hand, automatic data mining and data analytics have made tremendous progress in the past decade. Machine learning, particularly deep learning, has become the mainstream analytics method in most big data analysis problems. The effective integration of visualization and machine learning / data mining is a new challenge in big data research.

Machine learning algorithms such as neural networks and support vector machines use data to build computational models that are representations of nonlinear surfaces in high dimensional space. The trained models can then be used for analysis tasks such as classifications, regressions and predictions. Recent progress in deep learning has further empowered machine learning as an effective approach to a large set of big data analysis problems. As an automatic method, machine learning algorithms act mostly as a black box, i.e. the users have very little information about how and why the algorithm work or fail. The underlying machine learning models are also designed primarily for the convenience of learning from data, but they are not easy for the users to understand or interact with. Interactive machine learning aims to provide a mechanism through visualization to allow users to understand and interact with the learning process [2]. It has several important potential benefits.

1) Understanding

It is often difficult to improve the efficiency and performance of the algorithms without a clear understanding of how and why the different components work in machine learning algorithms. It is even more so in deep learning where there are large number of layers and interconnected components.

2) Knowledge Input

Human knowledge input can significantly improve the performance of machine learning algorithms, particularly in areas involving professional expertise such as medicine, science and engineering. Human instinct from visual perception can also outperform computer algorithms. Hence, it is important to develop a visualization supported user feedback platform to allow user input to the machine learning system in the form of feature selection, dimension reduction, parameter setting, or addition / revision of rules and associations.

3) Data Reduction

Automatic machine learning usually requires a large volume of data to train the underlying computational model. This strategy sometimes is not realistic for applications in which data collection, labelling or processing is very expensive or difficult (for example, in clinical trials). Interactive visualization of the machine learning process allows the user to iteratively select the most critical and useful subset of data to be added to the training process so that the model building process is more data efficient (Figure 1). This is also our primary focus in this paper.

Our goal is to develop a visualization supported user interaction platform in a machine learning environment such that the user can observe the evolution and performance of the

internal structures of the model and provide feedback that may improve the efficiency of the algorithm or correct the direction of the model building process. Although the visualization platform we develop can be used to support “understanding” and “knowledge input” functions, we focus specifically in this paper on “data reduction”. In our approach, the interactive system will allow the user to identify potential areas (in some visual space) where additional data is needed to improve or correct the model (as shown in Figure 1). This way, only the necessary amount of data is used for learning a model.

We aim to solve a big data problem using a small data solution. In practice, this approach can not only save costs for data acquisitions / collections in applications such as clinical trials, medical analyses, and environmental studies, but also improve the efficiency and robustness of machine learning algorithms as the current somewhat brute-force approach (e.g. in deep learning) may not be necessary with smaller and higher quality data. To achieve this goal, we will need to overcome the following two challenges:

1. How to visualize the dynamics of a machine learning model is technically challenging. Previous works often depend on the specific machine learning algorithms. But in this paper, we will develop an approach and a general strategy that can be applied to most machine learning algorithms. In our test applications, support vector machines will be used as an example to demonstrate the effectiveness of this approach.
2. How to identify problematic areas from the visualization to revise the model, and how to efficiently and effectively provide user feedback to the algorithm are challenging. This is because machine learning features are often non-trivial properties of the data which cannot be easily used to pre-screen potential data collection target in real world applications.

In this paper, we will present a solution to these two challenges. Our approach will be tested on two practical applications with real world datasets.

In the following, we will first, in Section 2, discuss previous work related to interactive machine learning and other visual analytics solutions. The interactive visualization platform along with our general visualization and interaction techniques will be described in Section 3. The two application problems: handwriting recognition and cognitive score prediction using human brain data will be presented in Section 4. Conclusions and future work will be given in Section 5

II. Related Work

Although interactive machine learning has been previously proposed in the machine learning and AI communities [2, 3], applying visualization and visual analytics principles in interactive machine learning has only been an active research topic in recent years. Most of the existing studies focus on using visualization for better understanding of the machine learning algorithms. There have also been some recent works on using visual analytics for improving the performance of machine learning algorithms through better feature selection or parameter setting.

While there have been many literatures on using interactive visualization to directly accomplish analysis tasks such as classification and regression [4, 5], we will focus mostly on approaches that deal with some machine learning models [6]. Previous works on using visualization to help understand the machine learning processes are usually designed for specific types of algorithms, for example, support vector machines, neural networks, and deep learning neural networks.

Neural Networks received the most attention due to its “black box” nature of the learning model and the complexity of its internal components. Multi-dimensional visualization techniques such as scatterplot matrix have been used to depict the relationships between different components of the neural networks [7, 8]. Typically, a learned component is represented as a higher dimensional point. The 2D projections of these points in either principal component analysis (PCA) spaces or a multi-dimensional scaling (MDS) space can better reveal the relationships of these components that are not easily understood, such as clusters and outliers. Several techniques have applied graph visualization techniques to visualize the topological structures of the neural networks [9, 10, 11]. Visual attributes of the graph can be used to represent various properties of the neural network models and processes.

Several recent studies tackle specific challenges in the visualization of deep neural networks due to the large number of components, connections and layers. In [12], Liu et al. developed a visual analytics system, CNNVis, that helps machine learning experts understand deep convolutional neural networks by clustering the layers and neurons. Edge bundling is also used to reduce visual clutter. Techniques have also been developed to visualize the response of a deep neural network to a specific input in a real-time dynamic fashion [13, 14]. Observing the live activations that change in response to user input helps build valuable intuitions about how convnets work.

There are several literatures discussing visualization’s roles in support vector machines. In [15], visualization methods are used to provide access to the distance measure of each data point to the optimal hyperplane as well as the distribution of distance values in the feature space. In [16], multi-dimensional scaling technique is used to project high-dimensional data points and their clusters onto a two-dimensional map preserving the topologies of the original clusters as much as possible to preserve their support vector models.

Visualization and visual analytics methods have been proposed for the performance analysis of machine learning algorithms in different applications [17, 18, 19]. Interactive methods have also been proposed to improve the performance of machine learning algorithms through feature selection and optimization of parameter settings. Some general discussions are given in [6] and [20]. In [21], a visual analytics system for machine learning support called Prospector is described. Prospector supports model interpretability and actionable insights, and provides diagnostic capabilities that communicate interactively how features affect the prediction. In [22], a multigraph visualization method is proposed to select better features through an interactive process for the classification of brain networks. Other performance improvement methods include training sample selection and classifier tuning [23] and model manipulations by user knowledge [24, 25, 26].

The incremental visual data classification method proposed in [23] has some similarities conceptually to what we propose in this paper. In [23], neighbor joining tree is used to classify 2D image data. The model building process is done incrementally by adding additional images that are visually similar to the test samples that were misclassified. This approach puts a very heavy burden on the user as finding similar images by the user from a large image database or other sources is difficult and time-consuming. Our approach is a more general framework that works for all machine learning algorithms and all data types. It is designed to allow incremental addition of training data with any user defined characteristics (attributes) that are easy to identify and collect.

III. A Framework

In this section, we present a framework of interactive machine learning by visualization. The application of this framework to two test examples will be discussed in Section 4.

A. System Overview

Our goal is to develop a new interactive and iterative learning technique built on top of any machine learning algorithm so that the user can interact with the machine learning model dynamically to provide feedback to incrementally and iteratively improve the performance of the model. Although there can be many different forms of user feedback, such as knowledge input, features selection and parameters setting, in this paper we focus primarily on adding the optimal subset of training data samples such that the added training samples can provide maximal improvement of the model using minimum number of additional training points. Hence, the problem statement can be formulated as follows:

Let F be the feature space of a machine learning algorithm, $X = \{x_1, x_2, \dots, x_n\} \subset F$ be the starting training set, and $Y = \{y_1, y_2, \dots, y_m\} \subset F$ be an internal test set. We define U as user space of the same dataset containing some user defined attributes. These user defined attributes are selected based on two criteria: (1) they are part of the attributes of the original dataset; and (2) they can be used to identify data points (to be collected) easily. Let $Z = \{z_1, z_2, \dots, z_m\} \subset U$ be set Y represented in the user space U , and $M_0(y): F \rightarrow C$ be the learned model using the initial training set X , where C is the application value range (e.g. class labels or regression function values).

We want to find a set of k new data points (where k is a constant), $X' \subset F$, such that points in X' satisfy a set of user defined conditions of attribute values in U . These conditions in U is defined interactively from the visualization of the model and its test results on set Z in the user space U . The user's goal is to provide additional training samples such that the learned model $M_1(y)$ using training set $X \cup X' \subset F$ is an improved model over $M_0(y)$.

The above process can continue iteratively until the performance of the model is satisfactory or until the model can no longer be improved.

This framework can be summarized by the structural flowchart in Figure 2. At each iteration, a machine learning model is constructed using the current training set. The model will be tested on an internal test set. The visualization engine will then visualize the model

along with the labeled internal test results. Based on this visualization, the user can decide to add new samples in the areas where the model performed poorly. These new samples will be added to the current training set to enter the next iteration.

B. User Space and User Interactions

A critical idea in our interactive machine learning framework is the separation of feature space and user space. During each iteration of the learning process, the additional training data is often not readily available, and needs to be acquired separately using some easy to use attributes.

A machine learning algorithm learns a model based on the features of the training samples. These features are either precomputed by some dimension reduction methods (e.g. PCA) or selected through some feature selection algorithms. It is generally not feasible to obtain features of any data item before the data is collected. This is particularly true for complex datasets where the collection of each data item requires significant effort and cost. For example, in medical analysis, the collection of detailed medical and health data for each patient or a control individual is very expensive and time-consuming.

In our approach, a specific subset of conditions for data is identified through the interactive visualization process and targeted for collection. Thus, the attribute conditions for this subset of data need to be something that are easy to be used for the identification and collection of data. For this reason, we define *user space* as a data representation space containing attributes that can be used as the identifiers of the target data subset for iterative data collection. This also means that the interactive visualization also needs to be presented in this user space so that the user can interactively define the attribute conditions for additional data samples.

A *user space* is typically defined by the user based on the application needs. The attributes in the user space may contain:

- **Common attributes.** These include simple common characteristics of data that can be used to identify the data easily. For instance, in medical diagnosis applications, these may include common demographic information and behavior data such as age, gender, race, height, weight, social behavior, smoking habit, etc.
- **Special attributes.** These are attributes the analysts have special interests in. For example, in bioinformatics, certain group of genes or proteins may be of special interests to a particular research problem, and can be extracted from a large database.
- **Visual attributes.** Visual data such as images or shape data maybe directly visualized as part of the user space so that the user can visually identify similar shapes or images as new samples

Through the visualization of the model and the associated labels of the testing samples, the user can specify conditions for user space attributes to identify new training samples. This is done based on several different principles:

- **Model Smoothness.** The visualization will show the shape of the learned model at each iteration. Visual inspection of the shape of the model can reveal potential problem areas. For example, if the model is mostly smooth but is very fragmented in a certain region, it is possible that the learning process does not have sufficient data in that region.
- **Testing Errors.** Errors from the test samples can provide hints about areas where the model performs poorly. These may include misclassified samples and regression function errors. In areas with significant errors, new samples may be necessary to correct the model.
- **Data Distribution.** There may be a lack of training data in some area in the user space. This can affect the model's accuracy and reliability. For example, a medical data analysis problem may lack sufficient data from older Asian female patients. To show this type of potential issues, the visualization system will need to draw not only the test samples, but also the training samples within the user space

C. The Visualization Platform

The visualization platform in our interactive machine learning framework serves as the user interface to support user interaction and the visualization of data and the model.

Although there are many different visualization techniques for multi-dimensional data [27], we choose scatterplot matrix as our main visualization tool as it provides the best interaction support and flexibility. We also choose to use heatmap images to visualize the machine learning model within the scatterplots since it treats the machine learning model as a black box function and thus allows the approach to be machine learning algorithm independent.

Figure 3 shows a general configuration of the scatterplot visualization interface. The upper-right half of the matrix shows the feature space scatterplots, the lower-left half shows the user space scatterplots and the diagonal shows the errors of the corresponding feature space dimensions. Within each scatterplot sub-window, two types of visualizations will be displayed: (1) the data (training or testing data points); (2) the current learned model. Each of the 2D sub-windows can also be enlarged for detailed viewing and interaction. In principle, the dimensionality of the feature space and the dimensionality of the user space are not necessarily the same. But for convenience, we may select the same number of features and user space attributes to visualize in this scatterplot matrix. It is certainly not hard to use different numbers of variables in these two spaces.

The primary challenge in this visualization strategy is the visualization of a machine learning model in a 2D subspace of the feature space or user space. A heatmap image filling approach will be used to visualize the model. Each pixel of the 2D sub-window will be sampled against the model function, and the result will be color-coded to generate a heatmap-like image. An example is shown in Figure 4 for a 3-class classification model.

Let the machine learning model be a function over the feature space, $M(Y): F \rightarrow C$, where F is the feature space and C is the range of the model function. The projection of the model in

a 2D subspace is, however, not well defined, and hard to visualize and understand. A better way to understand and visualize the model in a 2D subspace is to draw a cross-section surface (over the 2D subspace) of the model function that passes through all training points. Mathematically, this is equivalent to the following:

For a pixel point $P = (a, b)$ in a 2D subspace where a and b are either two feature values or two user space attributes, compute $M(y)$, where the feature vector $y \in F$ at P is calculated by interpolating the feature vectors of the training samples on this 2D subspace.

Any 2D scattered data interpolation algorithm can be used here to interpolate the feature vectors. In our implementation, since we need to interpolate all pixels in a 2D sub-window, a triangulation-based interpolation method is more efficient as the triangulated interpolants only need to be constructed once. The training data samples are triangulated by Delaunay triangulation first. A piecewise smooth cubic Bezier spline interpolant is constructed over the triangulation using a Clough-Tocher scheme [28]. An alternative method is to apply piecewise linear interpolation over the triangular mesh. But the cubic interpolation provides better smoothness. Please note that this interpolation scheme interpolates only the feature vectors, which will then be inputted to the model function to generate model output values for color coding.

Figure 5 shows two types of cross-sections. For simplicity of illustration, we use a 1D analog to the 2D cross-sections. So, the sample points on a 1D axis in the figure should be understood as the sampling points on a 2D scatterplot sub-window. Here (f_1, f_2) is the feature space. C is the model value, U is a 1D subspace of the user space. P_1 to P_5 are the training samples we use for interpolation. In Figure 5(a), f_1 axis is a subspace we use to visualize the model in the feature space. In Figure 5(b), U is a subspace we use to visualize the model in the user space. In this figure, we assume U is a linear combination (rotation) of the feature dimensions. But U sometimes can be wholly or partially independent of the features. In that case, interpolation will just simply be done within the user space similarly as in Figure 5(a). Since we have many 2D sub-windows in the scatterplot matrix, the combinations of these cross-sections provide a cumulative visual display of the model function at every iteration of the learning process.

IV. Test Applications

The framework described in the last section has been implemented using Python and a Python 2D plotting library: Matplotlib. In this section, we will apply this framework to two different types of test applications: handwriting recognition (classification) and human cognitive score prediction (regression) using real world datasets.

A. Handwriting Recognition

In this case study, we apply our interactive machine learning approach to the classification (recognition) of handwriting digits. The well-known publicly available MNIST handwritten digits dataset from the MNIST database [29] is used. For better illustration effect, we will narrow the recognition scope to four classes of digits: 0s, 1s, 2s, and 3s. For these 4 classes,

we have 24673 training points and 4159 testing points. We also picked 123 points (0.5%) out of the training set for internal test to guide the interactive process.

Each original data point contains a fixed sized 2D image. Principal Component Analysis (PCA) is applied to the pixel arrays of these images. The top 10 principal components are used as feature vector in a Support Vector Machines classification algorithm. This SVM process is very similar to a typical face recognition system [30].

In this application, the feature space is the principal component space. If we only use the top four features (PCs) for visualization and interaction, we will have a 4 by 4 scatterplot matrix. Since this is an image dataset, the user space contains the original pixels. It is easier to simply display small icons of some of the original images within the user space scatterplots. These icons can be enlarged when clicked by the user.

Figure 6 shows an interface for this interactive session with four features. The scatterplot matrix is symmetric, but the lower-left sub-windows show icons of some of the original images, which serve as a user space. Each diagonal sub-window shows a histogram of the distribution of the misclassified points in each feature dimension. It is certainly possible to use the shape information of the misclassified points to retrieve similar new samples from the large database, which can perhaps be automated. In our experiment, we focus only on interactive operations. New samples are added in areas where there are too many misclassified samples or the classification boundaries appear fragmented. The process started with only 10 training samples. In each iteration, 5 new samples are added at an area clicked by the user.

Figure 7 shows a performance chart for this experiment. The orange line represents the performance using randomly selected samples, and the blue line represents the performance using interactively selected samples. Since this problem is relatively easy, the performance curve converges quickly after about 250 points. But the blue line reaches the near-peak performance much earlier at about 75 points. Figure 8 shows a sequence of the interactions that led to iterative model improvement.

B. Human Cognitive Score Prediction

Understanding the structural basis of human cognition is a fundamental problem in brain science. Many studies have been performed to predict the cognitive outcomes from measures captured by Magnetic Resonance Imaging (MRI) scans of the brain [31–34]. In this case study, we apply our interactive machine learning approach to the prediction of cognitive performance using MRI data coupled with relevant demographics and behavior information.

The data studied in this work were downloaded from the Human Connectome Project (HCP) database [35–37]. HCP is a major NIH-sponsored endeavor that has acquired and published brain connectivity data plus other neuroimaging, behavioral, and genetic data from 1200 healthy young adults. Its goal is to build human brain network map (i.e., connectome) to better understand the anatomical and functional connectivity in relation to cognitive and behavior outcomes within the healthy human brain.

There are four types of attributes from the HCP database: (1) the Mini-Mental State Examination (MMSE) score, which is the cognitive outcome studied in this work; (2) structural MRI measures, including volume measures and cortical thickness measures of regions of interests generated by the FreeSurfer software tool [38]; (3) demographical measures such as age, race, weight, height, BMI, etc.; and (4) behavioral measures. Our computational task is to predict the MMSE score using the MRI, demographical and behavior measures.

A total of 1177 subjects with complete cognitive, imaging, demographical, and behavior information were included in our study. 589 subjects are used as test set and 588 subjects are used as training set. We selected 5% (about 30 samples) of this training set as internal test set to guide the user interaction. A principal component analysis (PCA) is used for feature selection. A support vector regression (SVR) technique is applied to the top 10 principal components (PCs) to obtain a regression model in each iteration for the prediction the MMSE scores [39, 40]. The predicted MMSE scores are then color coded to generate the heatmap images in the 2D sub-windows. The iteration starts with 10 initial training samples. In each iteration, 5 new samples are added at an area clicked by the user.

Figure 9 shows a screen shot of the interface. The top four principal components are used as the features in the scatterplots. The user space attributes used in this visualization include the patient's weight, height, age, and body mass index (BMI). The diagonal sub-windows show the histograms of the distributions of the regression errors for the individual feature dimensions. We again only focus on adding new samples in areas where there are too many mismatches of colors between the model and the samples or the regression heatmap appears too fragmented. In practice, experts may also use other professional knowledge to add samples that relate to a particular hypothesis. From Figure 9, we can see that the scores are very flat in most of the regions but can change quite drastically within some isolated small region.

Figure 10 shows the performance chart for this iteratively built model. The orange line represents the results using randomly selected samples, and the blue line represents the results using interactively selected samples. We first applied the support vector regression using the entire 588 training set. The resulting mean error is 0.8. The chart shows that the interactive model converge quickly to the optimal performance (0.8 error) after about 80 training samples while the randomly selected training samples struggle to converge. Figure 11 shows a sequence of interactions that led to iterative model improvement.

Conclusions

We have presented a general framework for a visualization supported interactive machine learning approach, and have tested the framework on two different types of application problems using real world datasets: handwriting recognition and human cognitive score prediction. The experiments show that interactively selected training samples can reach high performance quicker than randomly selected samples. This approach provides a new way to train a machine learning model using a small set of training samples. since human knowledge and perceptual instincts are used in the selection of the training samples, this

approach is potentially smarter and more efficient than traditional “big data” solutions. It is particularly useful for applications where high quality “big data” is not readily available or if the collection and labeling of the data is too expensive (e.g. in some biomedical data analysis applications). on the other hand, since this approach requires human in the learning loop, it may not be suitable for applications that require total automation (e.g. in real time robot vision).

Although this paper focuses on the “small data” solution, the visual analytics framework proposed here can be applied to other types of interactive machine learning problems such as human knowledge integration and model optimization. In the future, we would like to explore new solutions to interactive machine learning with knowledge (e.g. rules and constraints) input, parameter settings and other model optimization functions. We would also like to investigate ways to automatically evaluate the models so that the additional samples can be added automatically.

Acknowledgments

This work is supported by the NIH (NIBIB) grant: R01 EB022574.

References

- [1]. Thomas J, Cook K: Illuminating the Path: Research and Development Agenda for Visual Analytics. IEEE-Press (2005)
- [2]. Amershi S, Cakmak M, Knox WB and Kulesza T, 2014 Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4), pp.105–120.
- [3]. Ware M, Frank E, Holmes G, Hall M and Witten IH, 2001 Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3), pp.281–292.
- [4]. Paiva JG, Florian L, Pedrini H, Telles G, Minghim R, 2011 Improved similarity trees and their application to visual data classification. *IEEE TVCG* 17 (12), 2459–2468 [PubMed: 22034367]
- [5]. Xia Jing, Chen Wei, Hou Yumeng, Hu Wanqi, Huang Xinxin, Ebert David S. DimScanner: A Relation-based Visual Exploration Approach Towards Data Dimension Inspection. VAST 2016.
- [6]. Liu Shixia, Wang Xiting, Liu Mengchen, Zhu Jun. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics* 1 (2017) 48–56.
- [7]. Zahavy T, Ben-Zrihem N, Mannor S 2016 Graying the black box: Understanding dqns. In: *ICML* pp. 1899–1908.
- [8]. Rauber PE, Fadel S, Falcao A, Telea A, 2017 Visualizing the hidden activity of artificial neural networks. *IEEE TVCG* 23 (1), 101–110. [PubMed: 27875137]
- [9]. Tzeng FY, Ma KL 2005 Opening the black box - data driven visualization of neural networks. In: *IEEE Visualization*, pp. 383–390. 10.1109/VISUAL.2005.1532820.
- [10]. Harley AW, 2015 An interactive node-link visualization of convolutional neural networks. In: *International Symposium on Visual Computing Springer*, pp. 867–877.
- [11]. Streeter MJ, Ward MO, Alvarez SA, 2001 Nvis: An interactive visualization tool for neural networks.
- [12]. Liu M, Shi J, Li Z, Li C, Zhu JJH, Liu S, 2017 Towards better analysis of deep convolutional neural networks. *IEEE TVCG* 23 (1), 91–100. <http://dx.doi.org/10.1109/VISUAL.2017.7990001>. [PubMed: 27576252]
- [13]. Yosinski Jason, Clune Jeff, Nguyen Anh, Fuchs Thomas, and Lipson Hod. Understanding Neural Networks Through Deep Visualization. *ICML Workshop on Deep Learning*, 2015.
- [14]. Zintgraf Luisa M, Cohen Taco S, Adel Tameem, Welling Max. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. *International Conference on Learning Representations (ICLR)* 2017.

- [15]. Lim SeungJin. A Light-Weight Visualization Tool for Support Vector Machines. 25th International Workshop on Database and Expert Systems Applications, 2014.
- [16]. Hamel Lutz, Visualization of Support Vector Machines with Unsupervised Learning. IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, 2006.
- [17]. Ren D, Amershi S, Lee B, Suh J, Williams JD, 2017 Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE TVCG* 23 (1), 61–70. [PubMed: 27875134]
- [18]. Alsallakh B, Hanbury A, Hauser H, Miksch S, Rauber A, 2014 Visual methods for analyzing probabilistic classification data. *IEEE TVCG* 20 (12), 1703–1712. [PubMed: 26356884]
- [19]. Chuang J, Gupta S, Manning CD, Heer J 2013 Topic model diagnostics: Assessing domain relevance via topical alignment. In: *ICML*, pp. 612–620.
- [20]. Krause J, Perer A, Bertini E. Using Visual Analytics to Interpret Predictive Machine Learning Models. *ICML Workshop on Human Interpretability in Machine Learning*, 2016.
- [21]. Krause Josua, Perer Adam, and Ng Kenney. Interacting with predictions: Visual inspection of black-box machine learning models. *ACM CHI 2016*, 2016.
- [22]. Wang J; Fang S; Li H; Goni J; Saykin AJ; Shen L Multigraph Visualization for Feature Classification of Brain Network Data. *EuroVis Workshop on Visual Analytics (EuroVA)*, pp.61–65, 2016.
- [23]. Paiva JGS, Schwartz WR, Pedrini H, Minghim R, 2015 An approach to supporting incremental visual data classification. *IEEE TVCG* 21 (1), 4–17. [PubMed: 26357017]
- [24]. Liu M, Liu S, Zhu X, Liao Q, Wei F, Pan S, 2016 An uncertainty-aware approach for exploratory microblog retrieval. *IEEE TVCG* 22 (1), 250–259. [PubMed: 26529705]
- [25]. Choo J, Lee C, Reddy CK, Park H, 2013 Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE TVCG* 19 (12) 1992–2001 [PubMed: 24051765]
- [26]. Wang X, Liu S, Liu J, Chen J, Zhu J, Guo B, 2016 TopicPanorama: A full picture of relevant topics. *IEEE TVCG* 22 (12), 2508–2521. [PubMed: 26761818]
- [27]. Etemadpour Ronak, Linsen Lars, Paiva Jose Gustavo, Crick Christopher, Forbes Angus Graeme. Choosing Visualization Techniques for Multidimensional Data Projection Tasks: A Guideline with Examples. *VISIGRAPP 2015: Computer Vision, Imaging and Computer Graphics Theory and Applications*. pp. 166–186, 2015.
- [28]. Mann Stephen. Cubic precision Clough-Tocher interpolation. *Computer Aided Geometric Design*. Volume 16, Issue 2, 2 1999, Pages 85–88.
- [29]. LeCun Y, Bottou L, Bengio Y and Haffner P: Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE*, 86(11):2278–2324, 11 1998.
- [30]. Turk M; Pentland A (1991). “Face recognition using eigenfaces”. *Proc. IEEE Conference on Computer Vision and Pattern Recognition* pp. 586–591.
- [31]. Wan J, et al., Identifying the neuroanatomical basis of cognitive impairment in Alzheimer’s disease by correlation- and nonlinearity- aware sparse Bayesian learning. *IEEE Trans Med Imaging*, 2014 33(7): p. 1475–87. [PubMed: 24710828]
- [32]. Wan J, et al., Sparse Bayesian Multi-Task Learning for Predicting Cognitive Outcomes from Neuroimaging Measures in Alzheimer’s Disease. 2012 *Ieee Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012: p. 940–947.
- [33]. Wang H, et al., Sparse Multi-Task Regression and Feature Selection to Identify Brain Imaging Predictors for Memory Performance. *Proc IEEE Int Conf Comput Vis*, 2011: p. 557–562.
- [34]. Yan J, et al., Cortical surface biomarkers for predicting cognitive outcomes using group l2, l1 norm. *Neurobiol Aging*, 2015 36 Suppl 1: p. S185–93. [PubMed: 25444599]
- [35]. Van Essen DC, et al., The WU-Minn Human Connectome Project: an overview. *Neuroimage*, 2013 80: p. 62–79. [PubMed: 23684880]
- [36]. Marcus DS, et al., Human Connectome Project informatics: quality control, database services, and data visualization. *Neuroimage*, 2013 80: p. 202–19. [PubMed: 23707591]
- [37]. Glasser MF, et al., The minimal preprocessing pipelines for the Human Connectome Project. *Neuroimage*, 2013 80: p. 105–24. [PubMed: 23668970]

- [38]. Dale AM, Fischl B, and Sereno MI, Cortical surface-based analysis. I. Segmentation and surface reconstruction. *Neuroimage*, 1999 9(2): p. 179–94. [PubMed: 9931268]
- [39]. Drucker H, Burges CJC, Kaufman L, Smola A, and Vapnik V Support vector regression machines In Mozer MC, Jordan MI, and Petsche T, editors, *Advances in Neural Information Processing Systems 9*, pages 155–161, Cambridge, MA, 1997 MIT Press.
- [40]. Stitson M, Gammernan A, Vapnik V, Vovk V, Watkins C, and Weston J Support vector regression with ANOVA decomposition kernels In Scholkopf B, Burges CJC, and Smola AJ, editors, *Advances in Kernel Methods—Support Vector Learning*, pages 285–292, Cambridge, MA, 1999 MIT Press.</References>



Fig. 1. Ail example of iterative model improvement by interactively adding new samples.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

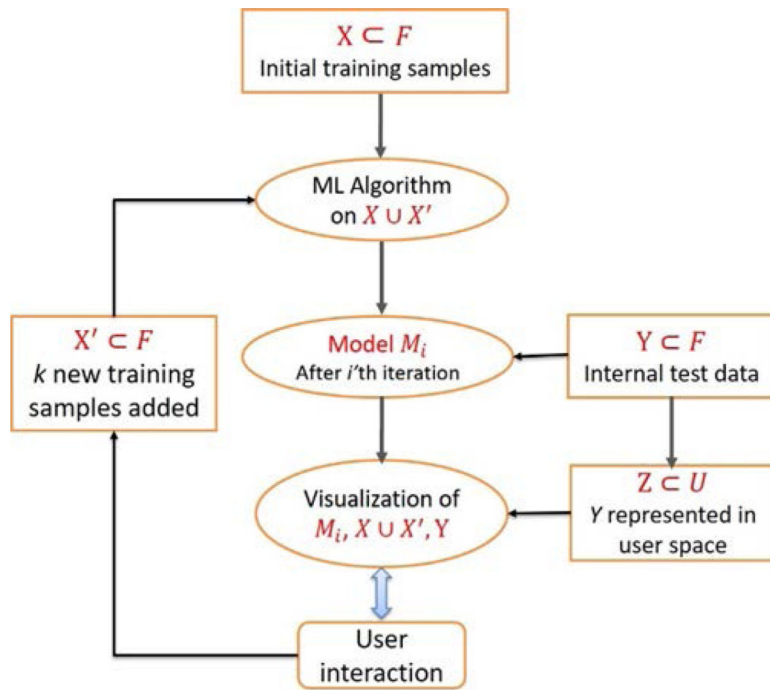


Fig. 2.
A structural flowchart of the interactive machine learning system.

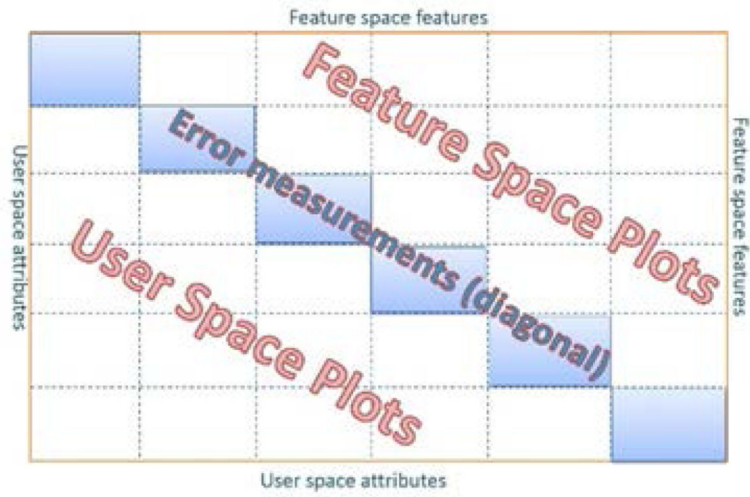


Fig. 3.
A configuration of the scatterplot matrix visualization.

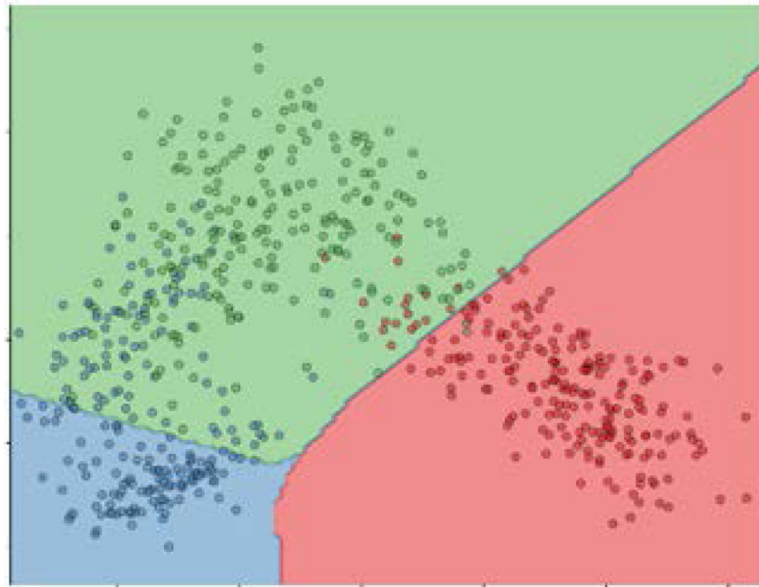


Fig. 4.
Model visualization example.

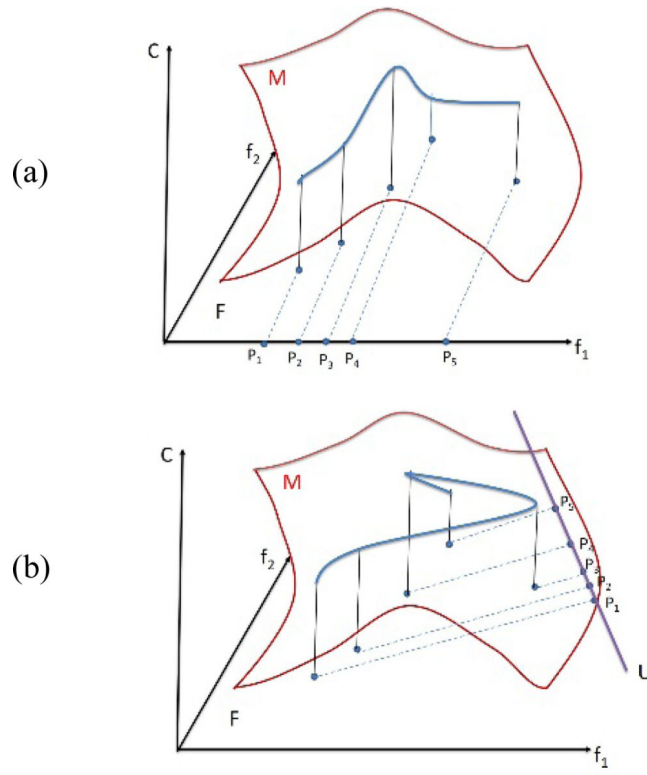


Fig. 5. 1D illustrations of model visualization as cross-sections (a) in feature space and (b) in user space.

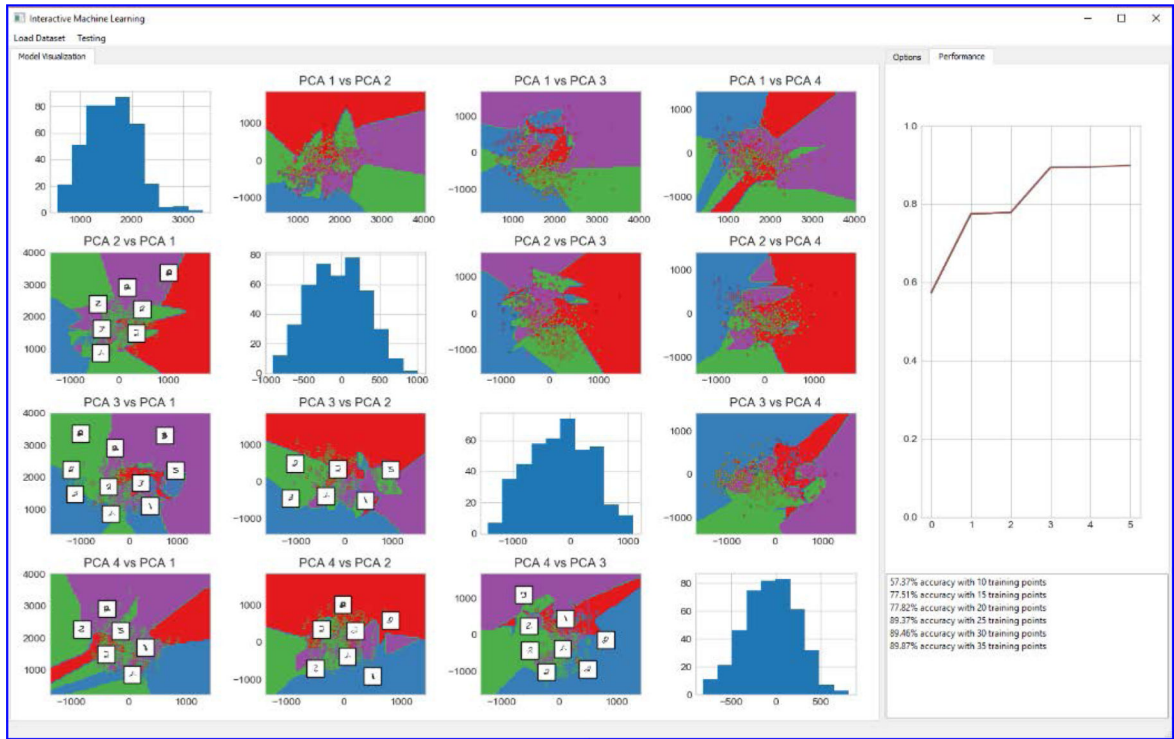


Fig. 6. Interactive machine learning interface for 4-class handwriting classification

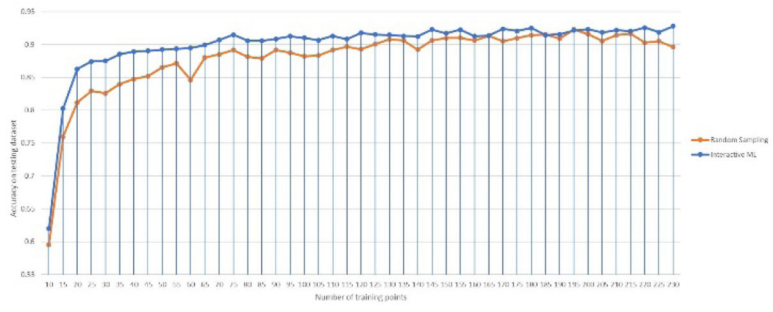


Fig. 7. Performance chart for handwriting classification experiment.

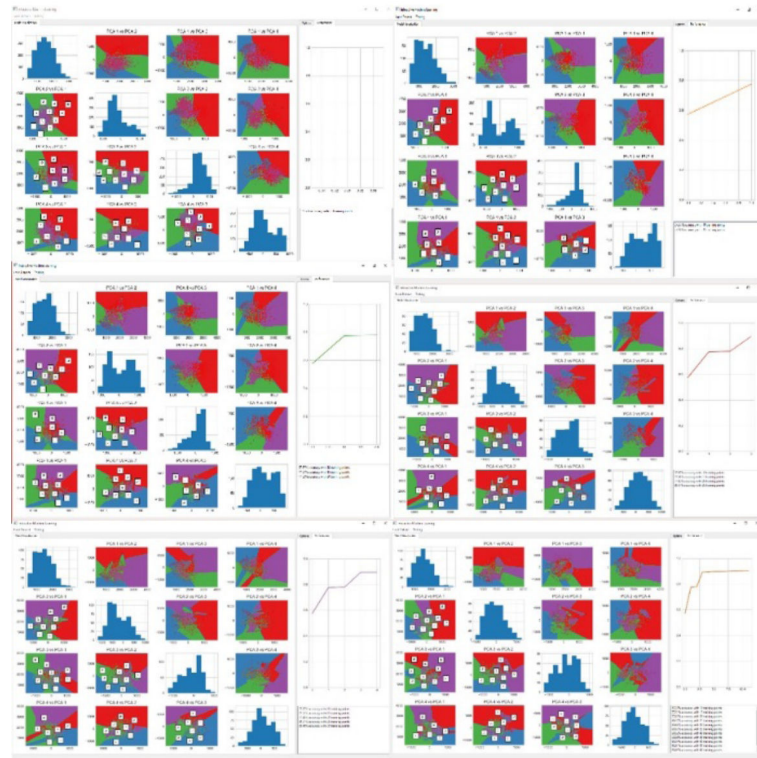


Fig. 8.
A sequence of model improvement iterations.

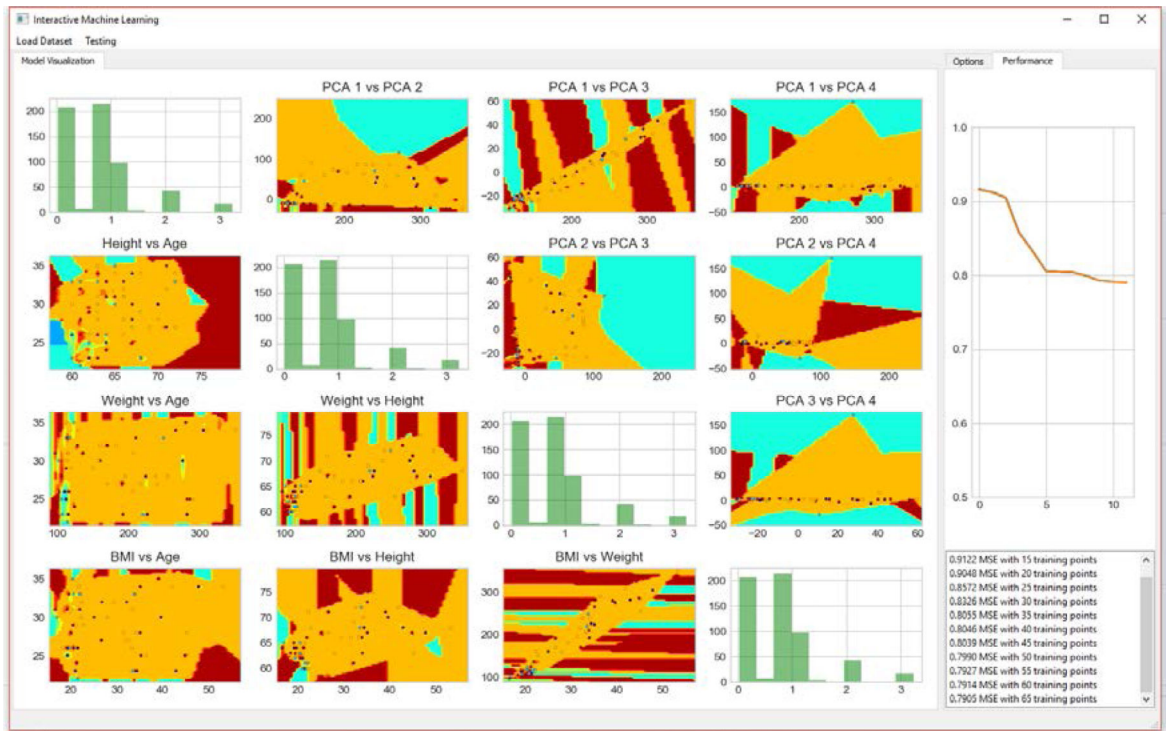


Fig. 9. Interactive machine learning interface for human brain data regression.

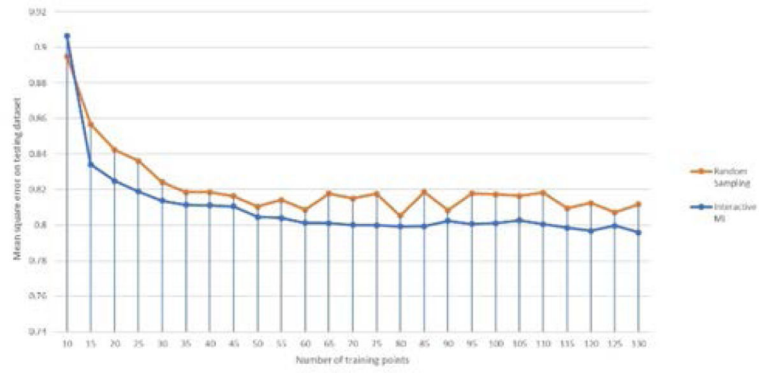


Fig. 10.
Performance chart for brain data regression.

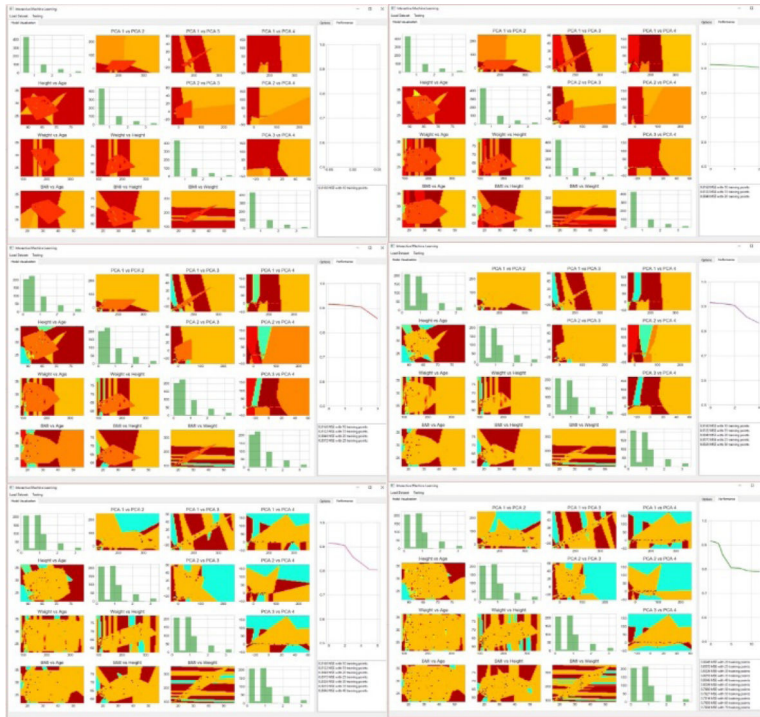


Fig. 11.
A sequence of model improvement iterations for MMSE score prediction.