An Analysis of Machine- and Human-Analytics in Classification

Gary K. L. Tam, Vivek Kothari, and Min Chen, *Member, IEEE*

Abstract—In this work, we present a study that traces the technical and cognitive processes in two visual analytics applications to a common theoretic model of soft knowledge that may be added into a visual analytics process for constructing a decision-tree model. Both case studies involved the development of classification models based on the "bag of features" approach. Both compared a visual analytics approach using parallel coordinates with a machine-learning approach using information theory. Both found that the visual analytics approach had some advantages over the machine learning approach, especially when sparse datasets were used as the ground truth. We examine various possible factors that may have contributed to such advantages, and collect empirical evidence for supporting the observation and reasoning of these factors. We propose an information-theoretic model as a common theoretic basis to explain the phenomena exhibited in these two case studies. Together we provide interconnected empirical and theoretical evidence to support the usefulness of visual analytics.

Index Terms—Visual analytics, classification, decision tree, model, facial expression, visualization image, information theory.

1 Introduction

Albert Einstein said: "In theory, theory and practice are the same. In practice, they are not." Following the spirit of his philosophical insight, we present an investigation into two visual analytics case studies, where human-centric processes were shown more advantageous than machine-centric processes. Both case studies involved the development of decision-tree models for classifying imagery data, and both featured information-theoretic measures in machine-centric processes. As information theory has been shown to be able to explain many phenomena in visualization [12], naturally, one would like to pose a question: can information theory explain the advantageous factors exhibited in the two case studies? In other words, could the two different instances of practice exemplify the same theory?

In the first case study, which was previously reported in [47], a visual analytics process was used to construct a decision-tree model for classifying four types of expressions (i.e., *anger*, *surprise*, *sadness* and *smile*) featured in facial videos. For each video, 14 time series representing different temporal facial features were first extracted from imagery data. For each time series, 23 quantitative measures were then obtained using different analytical metrics. These resulted in 14×23 attribute values per video. Tam *et al.* used parallel coordinates to select a small subset from the 14×23 attributes, and organized them into a decision tree that defines a model for classifying these expressions. They compared their model with decision trees resulted from machine learning, and found that human-constructed model was better. We briefly summarize this case study [47] in Section 4.

The second case study is a new development, where decision-tree models were constructed for classifying four different types of visual representations, namely, *bubble charts*, *treemaps*, *parallel coordinates* and *bar graphs*. In this study, visual analytics and machine-learning processes were running in parallel. A variety of image processing techniques were used to extract 222 features from visualization images. Information theory provided a measure of information gain that was used in both the human- and machine-centric approaches for constructing decision trees. The results showed that visual analytics processes performed better than machine-learning processes when sparse and skewed datasets were used for training, while both approaches performed similarly with a dense dataset. As this study has not yet been reported, we describe it with more details in Section 5.

Being the main researchers involved in the two case studies, we first conducted several brainstorming sessions to tease out various possible factors that might explain the advantages of the visual analytics approach. At the same time, we re-examined various computational data, and human decisions in the two case studies. We identified a number of factors suggesting that the human-centric approach benefitted from "soft knowledge", i.e., additional knowledge neither well-defined nor available to the machine-centric approach. The empirical observation and analysis of these factors are detailed in Section 6.

In order to bring machine-centric and human-centric processes into a common theoretic framework, we used information theory to model the factors identified in the empirical observation and analysis. In particular, we estimated the quantities of the Shannon information [14,45] available to both machine- and human-centric processes for constructing decision trees. This allowed us to reason the advantages of visual analytics workflows in these two case studies. This theoretical analysis is detailed in Section 7.

As constructing a decision tree is a form of model development, the human-centric processes in the two case studies are Model-developmental Visualization (Level 4) [11]. This work can inform model-developers about the necessity and means for involving humans in the loop. Our contributions include:

- We propose a thesis that information-theory can explain some common phenomena in visual analytics, where the development of data-driven models can benefit from the soft knowledge of model-developers. We outline an information-theoretic basis that can explain the merits exhibited in the two case studies in a quantitative manner, and provide a theoretical evidence to support the usefulness of visual analytics.
- We investigate into the relative merits of visual analytics and machine learning in two application case studies, where the usefulness of visual analytics was demonstrated by better decision trees constructed in human-centric approaches, yet the reasons were never comprehensively analyzed or well-understood.

2 Related Work

Many analytical problems, e.g., object recognition, outlier detection, and fault diagnosis, are *classification problems*. A solution to such a problem is referred to as a *classifier* or a *model*. One way to construct a classifier automatically is *supervised learning*. Given a training dataset consisting of multivariate data objects and their corresponding class labels, the goal is to infer, from the training dataset, a classifier that is able to predict the class label of any previously unseen data object. In the context of classification, we refer to those construction processes that are fully automated as "machine-centric", and those that involve humans in making indispensable decisions as "human-centric".

- *Gary Tam is with Swansea University. E-mail: k.l.tam@swansea.ac.uk.*
- *Vivek Kothari is with University of Oxford. E-mail: vivekoxford5@gmail.com.*
- *Min Chen is with University of Oxford. E-mail: min.chen@oerc.ox.ac.uk.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.
Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx/

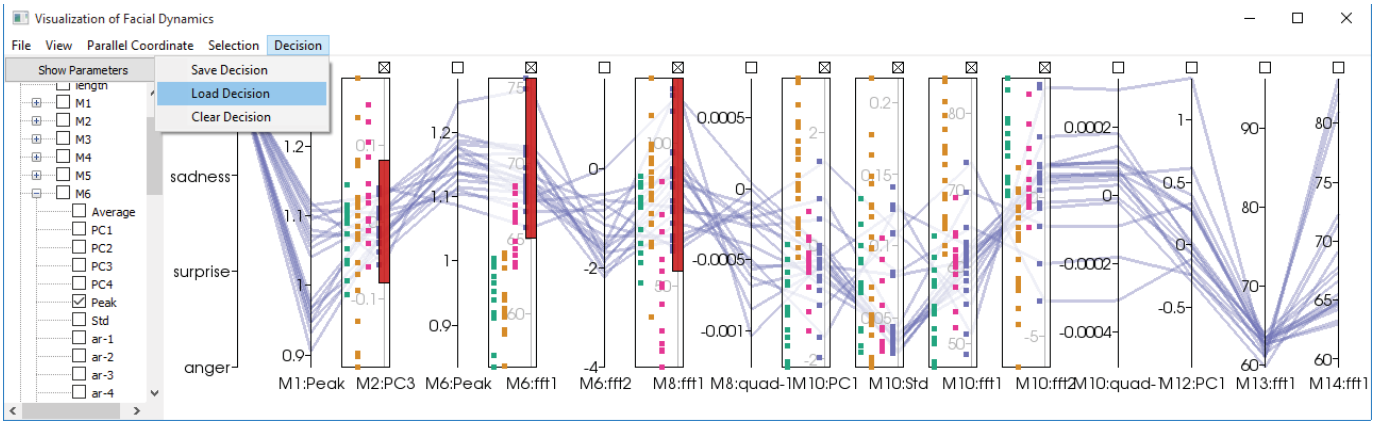


Fig. 1: A parallel coordinate plot used to aid model-developers in constructing a decision tree for classifying facial expressions. On the left, a model-developer can select facial features and time series attributes to investigate. On the main panel, each polyline represents a video and its attribute values are shown on the axes. The first axis indicates the classification labels. Scatterplots show the distributions of values and labels. Brushing is used to create and highlight decisions. The above example shows the three decisions used to obtain the *smile* classification.

2.1 Machine-centric Classification

There are many machine-centric techniques for constructing classifiers. Examples include multi-layered neural network, support vector machine, linear discriminant analysis, decision tree, Bayesian inference, and ensemble methods such as bagging, boosting and random forest [7,31,40]. Each technique can be highly successful in some scenarios where the structural assumptions of a model match the underlying relationship between data objects and class labels. There were several quantitative comparisons across some of these techniques, e.g., [2, 9, 10, 22].

Building highly accurate classifiers using machine-centric techniques is still a challenge in practice despite their extensive deployment. In general, machine-learning approaches are prone to (i) the curse of dimensionality where an exponentially large amount of data (and time) is theoretically required to train a model using high-dimensional data objects [7], (ii) the mismatch between the assumed model structure and the classification problem, leading to under- or over-fitting [38], and (iii) the lack of reliable training data, which may be due to poor feature separability, biases, noises and outliers in the data [18]. More often, the successful uses of machine learning require domain experts' insight about the structure of a model, sophisticated feature engineering, and non-trivial manual labelling effort [17].

Another drawback of machine-centric techniques is that the construction process and the resulting classifier are mostly opaque, and their inner workings are hard to comprehend [3]. Recently, visualization techniques were used for model developers to monitor the automated learning processes for neural networks [50, 51]. In comparison with most supervised learning approaches, the structure of a decision tree is perhaps more amenable to human examination. Its implementations and extensions (e.g., boosted tree and random forest) have been compared favorably in a number of studies [10, 22]. The decision tree technique thus offers a relatively impartial platform for juxtaposing a machine-centric process and a human-centric process in constructing a classifier. This technique will be further detailed in Section 5.3.

2.2 Human-centric Classification

Recently Sacha et al. reviewed the role of visual analytics in machine learning, and outlined several open challenges [44]. Crouser et al. suggested that the complexity of human computation can be measured [15]. Visual analytics can provide model developers with an effective means to explore data points in a high-dimensional space (e.g., [13, 30]). In particular, visual analytics has enabled human-centric approaches to be deployed for problems that are traditionally addressed using fully automated machine learning. Examples include image and video classification [5, 25, 33], anomaly detection [34], forensic risk assessment [37], and activity recognition [43]. In active learning, visual analytics can reduce manual labelling effort by allowing users to explore the training data, judge the quality of a classifier being trained, and steer the learning process accordingly (e.g., [35]). For constructing decision classifiers, the uses of visual analytics methods largely fall into two categories.

In the first category, the primary model construction process is automatic. Model developers may contribute their knowledge before the primary process (e.g., in choosing features [36]), or to support during the evaluation of a resultant classifier after the primary process. Examples for the latter include Mani Matrix [28], Interactive Confusion Matrix [29] and Confusion Wheel [1]. These techniques convey the testing results in a matrix display, and allow model developers to refine the parameters in the trained classifier (e.g., a fitness function) interactively. Adjusting a learned classifier through trial and error requires a comprehensive understanding of the inner workings of the classifier, and the process often brings about surprises as well as frustrations [28].

In the second category, model developers construct a decision-tree classifier manually, allowing the direct integration of human knowledge in making various algorithmic decisions (e.g., choosing a node). Ankerst et al. used a "circular segments diagram" to visualize the training data and aid the selection of attributes and cuts [4]. Tam et al. used parallel coordinate plots with embedded scatterplots for the same purpose [47]. van den Elzen and van Wijk used confusion matrices, a set of infographics, and tree visualization to aid the construction of a decision tree [49]. In addition, they also incorporated automated machine learning for creating a subtree if it is desirable. Most of the human-centric techniques in this category deal with a few to dozens of attributes, except in [47] where over 300 attributes were considered in the construction process. In [47], the decision tree constructed was compared favorably with that constructed using a machine-learning system C4.5. The work subsequently prompted an in-depth study of six state-of-the-art machine-learning techniques for facial expression classification, and the insights from [47] were used to improve the performance of these machine-learning techniques [20].

3 Theoretic Proposition

Data processing inequality (DPI) is a fundamental bottleneck in data analysis pipelines [14]. Any machine-centric process in Section 2.1 will likely involve multiple processing steps, and thereby will feature information loss (i.e., a phenomenon of DPI). Chen and Jänicke stated that interactive visualization could break the conditions of DPI [12]. The mechanisms include (i) human-computer interaction during a data analysis process, (ii) knowledge accumulated through training and experience that is not part of the data space, and (iii) knowledge about previous processing steps in addition to the immediately preceding one. Meanwhile, Chen and Golan considered "soft knowledge" featured in human-centric processes as undefined variables [11]. They

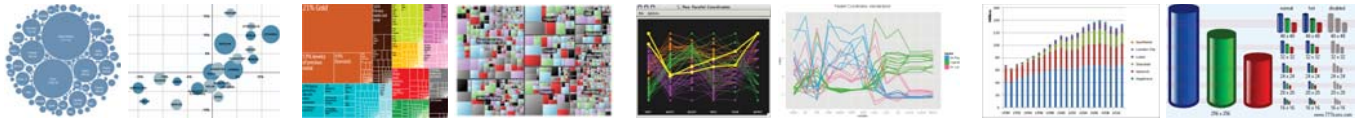


Fig. 2: Example visualization images in the training dataset. Four classes: bubble charts, treemaps, parallel coordinates, and bar graphs.

proposed a cost-benefit measure for taking such variables into account when optimizing a data analysis and visualization process. The mathematical reasoning in [11, 12] was based on information theory [45]. Since information theory provides a quantitative theoretical framework, it suggests that the quantities of "soft knowledge" used in human-centric processes (Section 2.2) can potentially be estimated.

In information theory, given a variable Z, the set of all valid values $\mathbb{Z}$ is referred to as an *alphabet*, and those valid values $z_i (i = 1,2,\ldots)$ as *letters* [11]. Given two alphabets $\mathbb{A} = \{a_1,\ldots,a_n\}$ and $\mathbb{B} = \{b_1,\ldots,b_m\}$, we can define a composite alphabet as $\mathbb{C} = \{(a_i,b_j)|a_i \in \mathbb{A} \text{ and } b_j \in \mathbb{B}\}$ where (a_i,b_j) is a valid pairing. In information theory, each letter in an alphabet is associated with a probability. In general, we should not assume that the probability of a composite letter (a_i,b_j) can always be mathematically derived from those of a_i and b_i, because of the likely mutual information between $\mathbb{A}$ and $\mathbb{B}$, and the semantically-complex and context-sensitive definition of validity.

Let M be a machine-centric process and H be a human-centric process. We can consider all possible inputs to process M as an alphabet $\mathbb{X}_M$, which may be a simple or a composite alphabet. If M has the access to all letters and their probability distribution in $\mathbb{X}_M$, it has the information measurable by Shannon entropy $\mathscr{H}(\mathbb{X})$. When the probability distribution of $\mathbb{X}_M$ is unknown, we can use the maximal entropy value of $\mathscr{H}(\mathbb{X})$, i.e., $\log_2||\mathbb{X}_M||$ bits as an estimation, where $||\mathbb{X}_M||$ is the number of letters in $\mathbb{X}_M$.

Assume that the human-centric process H performs a similar algorithm as M, H can thus access the same amount of information, i.e., $\mathbb{X}_M = \mathbb{X}_H$. For H, the cost of reading such information numerically may be much higher, while the cost of comprehending such information visually may be much lower. In addition, H may have access to "soft knowledge" as additional inputs. Some soft knowledge would be information uncaptured or underutilized by a similar machine-centric process. Other soft knowledge would be decisions that would not be part of the algorithm and would have to be made by a model developer. We will discuss these two types of soft knowledge in Section 7. Here we propose that such soft knowledge can also be estimated quantitatively using Shannon entropy. To demonstrate how this can be done numerically, we first describe two case studies in Sections 4 and 5, and then make empirical observation about the soft knowledge used in these two case studies in Section 6.

4 Case Study A: Facial Expression Classification

Analyzing facial videos and expressions (facial dynamics) is a challenging problem in computer vision. The progress to develop an accurate classifier has been slow, largely due to the huge video data space, the high-dimensional feature space, and the lack of sizeable and reliable training data. As reported in [47], visual analytics was used to help vision scientists explore a high-dimensional feature space. The raw dataset consists of 68 videos, each showing one of the four expressions, *anger*, *surprise*, *sadness* and *smile*. A video processing pipeline was devised for transforming each video to a feature vector. It includes steps for (i) applying groupwise registration to all frames of the videos, (ii) defining, tracking and extracting 14 facial measurements (e.g., movement of eye brows, changes of forehead textures, etc.), (iii) measuring time series using 6 types of feature transforms (e.g., PCA, Fourier transform, etc.) to obtain 23 features. Each video results in a feature vector with $14 \times 23 = 322$ attributes.

Parallel coordinates with embedded scatte rplots were then used to explore the feature space. Fig. 1 shows the interface of the visualization tool. On the left, a panel allows different attributes to be selected. On the main panel, the selected attributes are shown as parallel axes. Each connecting polyline represents one video and is colored according to its label. The distribution of all videos on each axis can be further explored with the embedded scatterplot. Brushing can be applied on multiple axes for selection. The selected axes and decision boundaries can be saved and reloaded using the pull-down menus. The tool supports exploratory visualization, observation of clusters and anomalies in the training data, analysis of correlation and orthogonality of axes, and judgment of separability of, and a potential split on each axis. It enables vision scientists to construct a decision-tree classifier iteratively, while observing classification results immediately through the parallel coordinates. The domain knowledge of the users played an important role in filling the gaps in the sparse data space, handling noise and uncertainty in the feature space, alleviating under- and over-fitting problems, and so on. More detailed observation about such knowledge will be made in Section 6.

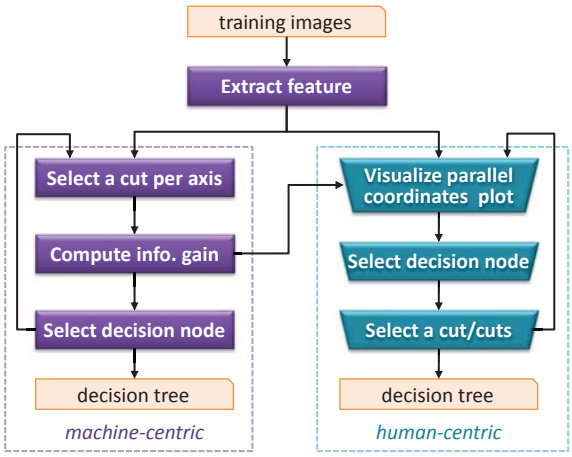


Fig. 3: The two pipelines, representing the machine- and human-centric approaches respectively, for constructing decision trees that can detect the visual representation featured in a visualization images.

5 Case Study B: Visualization Image Classification

Image classification is a classic problem in computer vision (e.g., [7,40]). This case study is concerned with a subdomain for classifying visualization images. As shown in Fig. 2, we considered four types of visual representations, and would like to construct a classifier that could automatically label a visualization image with its corresponding representation type. A technical solution for such a problem can be used in image search engines, spam filters, data mining (e.g., popularity ranking), ontology learning, and so on. In this case study, we compare two approaches to the construction of a classification model, namely *machine learning* and *visual analytics*. Similar to Case Study A in Section 4, we focus on decision trees as the underlying model structure. Fig. 3 illustrates the workflows of the two approaches.

5.1 Problem Statement and Dataset

Consider a collection of visualization images, and each image is labeled with its type, e.g., bubble chart, treemap, etc. We denote such a collection as a set of 2-tuples $\mathscr{I} = \{(I_1,\tau_1),(I_2,\tau_2),\ldots,(I_n,\tau_n)\}$ where I_i is an image, and τ_i is its type. The set is normally divided into a *training set* and a *testing set* such that $\mathscr{I} = \mathscr{I}_{train} \cup \mathscr{I}_{test}$ and $\mathscr{I}_{train} \cap \mathscr{I}_{test} = \emptyset$. A solution to a classification problem is thus to discover a suitable model $\mathbf{M}$ that takes an image I as the input, and predicts a type τ' as the output. The discovery process for an image classification model typically involves three stages.

Stage 1. A set of features $f_{1...m}$ are extracted from each image $I \in \mathscr{I}$. Different types of features usually require different feature extraction algorithms. Here we use $\mathbf{E}$ to denote these algorithms collectively, and it thus functionally maps an image to a feature vector as $\mathbf{f} = \mathbf{E}(I)$.

Stage 2. A model $\mathbf{M}$ is then constructed using the subset $\mathscr{I}_{train} \subset \mathscr{I}$ for training. As shown in Fig. 3, we compared two approaches for constructing such a model. (i) We used automated machine-learning processes to constructed a decision tree based on a collection of feature vectors extracted from images in $\mathscr{I}_{train}$. Two well-established machine-learning algorithms were used, and they are C4.5 [42] and CART [8]. (ii) We used parallel coordinates to aid a human-centric process for constructing a decision tree.

Stage 3. Given a constructed model $\mathbf{M}$, we evaluate its quality by using $\mathscr{I}_{test} \subset \mathscr{I}$ for testing. For each image $I \in \mathscr{I}_{test}$, we make pairwise comparison between the ground truth τ and the predicted label τ' generated by invoking the constructed model with extracted feature vector as $\mathbf{M}(\mathbf{E}(I))$. The mean accuracy for all images in $\mathscr{I}_{test}$ indicates the quality of the model.

The above three stages are detailed in the following four sections. The first two stages are illustrated in Fig. 3.

Dataset. For a number of well-structured subdomains of image classification, there are large datasets curated purposely as community resources for testing various solutions. These include COIL [39] and CIFAR-10/100 [32]. However, visualization images are very rare in these datasets. We thus collected a set of visualization images from the Internet. While these images are of varying quality, many contain various artifacts, and all suffer from some quality degeneration due to lossy compression. Four different classes of visualization images were chosen, namely *bubble charts*, *treemaps*, *parallel coordinates*, and *bar graphs*. We then chose an equal number of 49 JPEG images randomly from each class, avoiding any bias in the dataset (i.e., total $4 \times 49 = 196$ images). Two examples of each class is shown in Fig. 2.

5.2 Feature Extraction

An image feature f is a piece of information about an image I. There are many types of features, ranging from different statistical indicators about I or parts of I (e.g., its color histogram, Fourier spectral signature, Gabor texture descriptors) [21] to descriptions of patterns in I (e.g., edge detection, face detection, etc.) [40]. In image processing and computer vision, it is common to extract a set of features to form a feature vector $\{f_1, f_2, \ldots, f_m\} = \mathbf{f} = \mathbf{E}(I)$. Further algorithmic processing, such as image classification and clustering, can be formulated in such a feature space. This is referred to as the *bag of features* method [16], which is derived from the *bag of words* method in natural language processing and information retrieval [24]. In this case study, we extracted $m = 222$ features, which are briefly described below.

Image Metadata. We considered that the resolution, height, width and aspect ratio might contain information about the types of visual representations. For example, some images might be more likely to be rasterized at a higher resolution. Some might appear more commonly in a square shape than others. We purposely excluded the file name or keyword tags, as this might give the human-centric approach a significant advantage.

Background Properties. Maximizing the use of visual channel (e.g., space) is one of the key elements in visualization. We considered that the background of a visualization image might also contain some useful information for classification. We extracted two features: color and percentage of the background. In image processing, the background of an image is typically characterized by lower order moments. We first normalize an image to a square, compute the first 11 Zernike moments [48], and used their amplitudes and phases values.

Speeded-Up Robust Features (SURF). SURF is a local feature descriptor proposed by Bay *et al.* [6]. It forms the scale space of an image by progressively blurring the image with Gaussians, and detects interesting points from local extrema in the scale space. SURF then extracts histograms from the local region around the interesting points, and encodes them as a 64-D vector.

Texture & Artifact Distribution. Different visual representations may consist of different amount of detailed textures and artifacts. For example, treemaps may have large monochromatic areas, while its text labels may appear as small artifacts. Most bubble charts are expected to be full of small and/or large circles. Often artifacts may be highly local, concentrated at certain areas (e.g., near axes). The statistics of large and small scale features is therefore useful. We calculated the first four normalized moments of all Fourier coefficients (magnitudes and phases), and used moment invariants to characterize the global geometric distribution. Large coefficients (responses) at high frequencies imply that much detail exists across an image.

Geometric Structures. Geometric shapes (e.g., lines, curves, circles) are important visual channels in visualization. Some images may consist of circles (e.g., bubble charts), whilst some consists of many parallel lines (e.g., bar chart) or a more uniform distribution of different line orientations (e.g., parallel coordinate plots). We considered two kinds of geometric structures. First, we used Hough transform to detect the distribution of circular objects and computed the normalized moments of the distribution of radii and the moment invariants of the distribution of centers. Second, we used Gabor filters to describe the apportionment of lines and angles. We computed the algebraic moment invariants to preserve the spatial distribution of the Gabor filter responses at 3 scales and in 8 orientations.

5.3 Machine-centric Decision Tree Construction

As shown in the pipeline on the left in Fig. 3, once a feature vector $\mathbf{f} = \mathbf{E}(I)$ is obtained for every image $I \in \mathscr{I}_{train}$, machine learning can be deployed to construct a decision-tree model $\mathbf{M}$. The learning process assumes an underlying model structure, where a class label τ' can be predicted by evaluating a set of features in a certain order. In many cases, the learning is further restricted to binary partition of each feature, and thus the model space consists of only binary trees. Here, we used C4.5 [42] and CART [8] for the machine-centric approach.

Let $\mathbb{D}$ be the m-D feature space, and a feature vector $\mathbf{f}$ is thus an m-D point in $\mathbb{D}$. In our case, $m = 222$. Whenever a feature $f_x \in \mathbf{f}$ and a cut value v_x are selected by the learning process, $\mathbb{D}$ is partitioned into two subspaces, $\mathbb{D}_1$ and $\mathbb{D}_2$. The process continues recursively for each subspace by selecting another feature to partition. It usually terminates when the tree reaches a certain depth, or when all instances (images) in each subspace at a leaf node are (mostly) of the same class (visual representation). Tree pruning may be applied to reduce over-fitting.

In the process of learning a decision tree, the most important actions are to (i) select a feature $f_x \in \mathbf{f}$ for a decision node, and (ii) determine a cut v_x to split the numerical range of f_x. In an automated process, these two actions are usually performed as an integrated step. In ID3 [41], information theory is used to determine an optimal cut. Given a potential split of $\mathbb{D}$ into $\mathbb{D}_1$ and $\mathbb{D}_2$ with f_x and v_x, ID3 computes:

$$Gain(\mathbb{D}, \mathbb{D}_1, \mathbb{D}_2) = \mathscr{H}(\mathbb{D}) - \widehat{\mathscr{H}}(\mathbb{D}_1, \mathbb{D}_2) \tag{1}$$

where $\mathscr{H}(\mathbb{D})$ is the Shannon entropy of $\mathbb{D}$. In our case, this concerns the information contains in an alphabet of four letters, each corresponding to a type of visual representation (more details in Section 7). The probability distribution of $\mathbb{D}$ is estimated based on the numbers of instances belonging to each class. If all images in $\mathbb{D}$ have the same class label, $\mathscr{H}(\mathbb{D}) = 0$, there is no need to split $\mathbb{D}$. If $\mathbb{D}$ contains a very mixed set of image labels, the entropy will be high. Let $w_i = |\mathbb{D}_i|/|\mathbb{D}|$, which gives the relative size of a subspace in comparison with its parent space. $\widehat{\mathscr{H}}(\mathbb{D}_1, \mathbb{D}_2) = \sum_{i=1}^{2} w_i \mathscr{H}(\mathbb{D}_i)$ is the weighted average entropy of the potential split. The information gain $Gain()$ provides a way to evaluate whether the split results in a reduction of entropy.

C4.5 [42] evaluates a slightly different measure:

$$GainRatio(\mathbb{D}, \mathbb{D}_1, \mathbb{D}_2) = \frac{Gain(\mathbb{D}, \mathbb{D}_1, \mathbb{D}_2)}{Split(\mathbb{D}, \mathbb{D}_1, \mathbb{D}_2)} = \frac{Gain(\mathbb{D}, \mathbb{D}_1, \mathbb{D}_2)}{\sum_{i=1}^{2} w_i \log_2(w_i)} \tag{2}$$

CART [8] uses a probabilistic measure:

$$CART(\mathbb{D}, \mathbb{D}_1, \mathbb{D}_2) = 2 w_1 w_2 \sum_{j=1}^{k} \left| p(\tau_j | \mathbb{D}_1) - p(\tau_j | \mathbb{D}_2) \right| \tag{3}$$

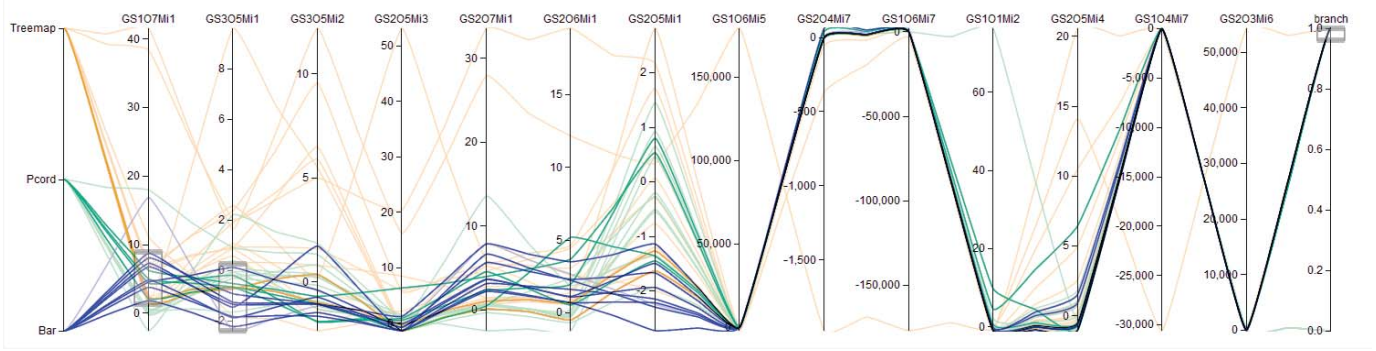


Fig. 4: A parallel coordinates plot used in the human-centric approach for image classification. Axes are ordered by *GainRatio*() in Eq. (2). Comparing the "best" ranked axis (GS107Mi1) and the second "best" (GS305Mi1), a human developer would consider that the second "best" has a few advantages (e.g., separability between green and blue lines).

where k is the number of classes, and $p()$ is the probability of class τ in a subspace. In our case, $k = 4$.

It has been shown that constructing an optimal binary decision tree is NP-complete [26] and an optimal general decision tree, NP-hard [23]. As features are normally defined by real values, different cut points v_x between two consecutive feature points on a feature axis f_x do not affect the above measures, but can affect the accuracy of the constructed model $\mathbf{M}$. Furthermore, when the number of instances is low, the estimation of entropy in ID3 and C4.5, and the probability in CART will be very unreliable. In our case, the dataset consists of 196 feature vectors in a 222-D space. Even for the first iteration, one could not expect to estimate the probability distribution for the whole feature space $\mathbb{D} = \mathbb{R}^{222}$ using 196 feature vectors. When the space was further divided into numerous combinations of $\mathbb{D}_1$ and $\mathbb{D}_2$, each based on a different coupling of an axis f_x and a cut v_x, all subspaces will contain much fewer feature vectors, and some could be down to a single digit. The poorer estimation of the probability distribution is, the less accurate estimation of the Shannon entropy is, and the worse selection of a decision node and a cut will be.

5.4 Human-centric Decision Tree Construction

The decision-tree construction process demands a huge amount of information in the 222-D feature space. Inspired by the success of [47], we conducted experiments to construct decision trees manually with the aid of visualization. We first tried several visualization techniques, including parallel coordinates, scatterplots, node-link tree visualization, and sunburst tree visualization. This allowed us to figure out how these known visualization techniques could aid a human-centric approach. We gained several initial observations.

Parallel Coordinates. Similar to [47], we used parallel coordinates to visualize the 222-D feature space, where each of the 196 feature vector $\mathbf{f}$ is a polyline connecting 222 axes. The polylines are color-coded according to its known class label τ. We use linear scaling with respect to the maximum and minimum values for each axis. We included interactions for (i) axis / intercept brushing and strumming selection for close inspection of a subset of feature vectors, and optional cut positions; (ii) Bezier curve bundling with sliders that control the curve smoothness and strength for observing similarity and diversity of a subset of feature vectors, and potential clusters determined by a group of nearby axes; and (iii) reordering axes for examining correlation and independence of a group of axes, and for prioritizing the effort for examining individual axes in detail. Fig. 4 shows one of the many parallel coordinates plots used in the experiments.

For selecting an axis and a cut using parallel coordinates, we found that one could handle a few axes at ease. However, inspecting a large number of axes visually is an arduous task, demanding a huge amount of concentration for visual reasoning, and cognitive effort for translating different visual patterns to qualitative judgements about promising axes and cut positions. Since we have C4.5 running log available, we used the *Gain*() in Eq. (1) and *GainRatio*() in Eq. (2) to prioritize axes for inspection. Users can switch between the two orderings. By using the axes-reordering interaction, one could further manually move promising axis to the front of the plot (or any user-designated region).

Scatterplots. When considering several optional cut positions, we found that the judgment about outliers was critical. We thus experimented with a common approach, where dimensionality reduction and scatterplot are coupled together to aid outlier detection. We used t-Distributed Stochastic Neighbor Embedding (t-SNE) for dimensionality reduction because of its computational performance, since dimensionality reduction had to support interactive inspection of the division of feature vectors at each iteration step. Two scatterplots were used to plot the projected 2-D feature points in the two potential subspaces determined by an axis f_x and a cut v_x.

However, we found it difficult to judge the potential outliers reliably using such scatterplots. We attributed the difficulties to the complexity of outlier detection. In a 222-D feature space, any of 196 feature vector was potentially an outlier. We thus resorted to human judgement of an outlier at an axis through manual inspection of the corresponding images, and its position on the axis.

Decision Tree Visualization. Naturally, one would like to see the partially constructed decision tree during a human-centric construction process. We experimented with two visual representations, a node-line tree with textual labels and a sunburst tree visualization. We found that the node-link tree offered a means of keeping a record about the axis and cut selected at each iteration, and the number of feature vectors in each subspace. We found that the sunburst tree visualization was useful for depicting errors during the testing stage. In general, their contribution to the complex decision as to which axis or cut position to choose was limited. We did, however, consider the possibility of designing a more advanced tree representation in the future that could support the explorative activities in the human-centric construction process. Nevertheless, such a design study would require a different project to accomplish and is beyond the scope of this work.

Visual Analytics for Decision Tree Construction Fig. 3 shows a human-centric pipeline (on the right) for decision tree construction, juxtaposing it with the machine-centric pipeline (on the left). Unlike the machine-centric approach, a model-developer normally selects an axis first before considering the cut positions in detail. In our case, the prioritization of axes inspection benefitted from the information-theoretic measures used in the machine-centric approach (i.e., *GainRatio*() from C4.5). These measures actually contain some information about cut positions, since the machine-centric approach has to evaluate optional cuts for each axis before evaluating all axes.

We constructed our decision trees primarily with the aid of parallel coordinates. The "drag-and-drop" mechanism for reordering axes was extensively used in the process to move prioritized axes closer together, and to inspect the proposed cuts by C4.5. We would typically inspected 5-20 axes before making a decision. We almost always adjusted a proposed cut position using brushing interaction, and in some cases, adding additional cut positions.

Table 1: Performance of the classifiers with a large dataset

	Accuracy	F-Measure	Kappa	AUC ROC
C4.5	68.33%	0.672	0.578	0.789
CART	68.33%	0.672	0.578	**0.796**
VA	**70.00%**	**0.693**	**0.600**	0.790

Table 2: Performance of the classifiers with a small dataset

	Accuracy	F-Measure	Kappa	AUC ROC
C4.5	62.50%	0.622	0.500	0.765
CART	66.18%	0.649	0.549	**0.809**
VA	**69.12%**	**0.691**	**0.588**	0.794

Table 3: Performance of the classifiers with a skewed dataset

	Accuracy	F-Measure	Kappa	AUC ROC
C4.5	61.67%	0.619	0.489	0.752
CART	51.67%	0.511	0.356	0.678
VA	**71.67%**	**0.713**	**0.622**	**0.826**

In addition, the system also allows a user to visualize the partial tree constructed after each iteration, and automatically save the constructed decision tree. There are also additional facilities for selecting the testing dataset randomly from the originally collected dataset, and for conducting the tests by running the decision-tree model automatically against a testing data set.

5.5 Experimental Results and Comparative Evaluation

To compare the performance of machine- and human-centric approaches for decision tree construction, we used several evaluation measures commonly used in the literature as advised in [46]. They are *mean accuracy*, *F-measure*, *Cohen's kappa coefficient* and the *AUC ROC* (area under the curve of receiver operating characteristics). Mean accuracy is a measure corresponds to the average accuracy of correctly predicted labels. F-measure takes the precision and recall into account (including the statistics of incorrectly predicted labels), and is a measure of relevancy of a classifier. Kappa coefficient compares the accuracy of the test classifier to the accuracy of a random classifier. ROC curve is a plot of the true positive rate and the false positive rate. The area under the curve measures the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance, and it is less sensitive to the underlying distribution of classes, compared to kappa coefficient. For all these measures, the higher the values indicate a stronger classifier performance.

We compared the human-centric approach in Section 5.4 against two algorithmic solutions for the machine-centric approach, namely C4.5 [42] and CART [8]. In particular, we conducted three experiments with different training datasets. Each dataset exemplifies a situation that is commonly encountered in real-world classification tasks.

Large Data Set. For machine-learning techniques, usually the more the data samples, the better the performance of the trained classifier. To simulate such a situation, the dataset described in Section 5 was randomly divided (0.3:0.7) into a test $\mathscr{I}_{test}$ and training set $\mathscr{I}_{training}$. Based on the same $\mathscr{I}_{training}$, three decision trees were constructed using C4.4, CART, and the human-centric approach (VA) respectively. They were then evaluated against $\mathscr{I}_{test}$. Table 1 shows the four measures of the experiment. Three measures indicate that VA was the best, and one measure is marginally in favor of CART.

Small Data Set. In many real-world applications, annotated training datasets with ground truth labels are not easily available. In this experiment, we used a small training set. We randomly divided the data (0.7:0.3) into $\mathscr{I}_{test}$ and $\mathscr{I}_{training}$. Table 2 summaries the results obtained, and it shows a similar pattern as Table 1, except the gaps between different methods were widened. In other words, the better performance of the human-centric approach became more obvious in this experiment. Our initial reasoning suggests that model developers may be more intelligent in generalizing their observations of a small dataset, spotting outliers and ignoring their effects [27].

Skewed Data Set. In the third experiment, we divided the data randomly (0.3:0.7) into a $\mathscr{I}_{test}$ and $\mathscr{I}_{training}$. We then removed 60% of bubble charts from the training set. This created a highly biased situation where there were much fewer samples for the bubble chart class than other three classes.

Table 3 summaries the results, and shows that the human-centric approach has a clear lead in all four measures. Our initial reasoning suggests that the human-centric approach may have benefitted from the ability of model developers to use their prior knowledge to fill in the gap caused by missing values [19]. Overall, we have found that the human-centric approach for decision-tree construction, with the aid of visualization and computer-based feature extraction and information-theoretic measures, performed better. The experiments have also shown that the gap between the two approaches narrows when the size of training dataset increases and biases are removed. In general, the human-centric approach seems to be more effective when conditions are less than ideal as in many real-world applications.

6 Empirical Observations

The work described in Section 4 was conducted by a team of seven researchers with expertise in computer vision, visual analytics, computer graphics, and machine learning. The human-centric decision tree was constructed by a researcher who was specialized in computer graphics and acquired the knowledge of computer vision and visual analytics during the project. The work described in Section 5 was carried out by two researchers with expertise in image processing and visual analytics. The three human-centric decision trees constructed by a researcher specialized in image processing with 8 months of experience in visual analytics at that time. As both case studies show the merits of the human-centric approach, it is curious as how visual analytics actually help. In this section, we summarize six empirical observations (**O1**-**O6**) made by the two model developers.

O1: Overview and Axis Distributions. As all features in the two case studies are defined as real values, it is not possible to evaluate all possible cut positions. Whilst the machine-centric approach can examine many cut positions on all axes and greedily pick the cut with the highest quality measure (Section 5.3), a human model developer usually first obtains a general overview of the data and identifies important axes with promising patterns (e.g., clusters, separability), before paying detailed attention to these axes. In **Case Study A** (facial expression classification), many features are clearly not useful (e.g., Fig. 5b), the model developer discards them quickly through visualization, and then focuses on features that appeared to be more useful (e.g., Fig. 5c). Further, apart from looking for cuts, a model developer also searches for axes with reliable clusters. In Fig. 1, for example, there are clear clusters for the highlighted lines (*smile*) on axes M2:PC3, M6:fft2 and M13:fft1. Visualization helps reduce the search space, especially when dealing with hundreds of features [47], and implicitly provides a quality measure of axes. As discussed in **O6** (domain knowledge), these axes were later confirmed to be useful. In **Case Study B** (visualization image classification), a different developer intuitively used the same top-down approach without any advice from the developer in **Case Study A**. The developer revealed that often the clusters were not perfectly separable from others, and human insight were drawn from various patterns approximately.

O2: General Agreement amongst Statistics. The three machine-learning algorithms discussed in Section 5.3 use only one single measure to evaluate a potential axis or cut. On the contrary, when presented with several statistical measures, a model developer is capable of evaluating and comparing several criteria at the same time. The agreement across different measures is often more important than a high value for a single criterion. For example, in **Case Study B**, there were occasions where one cut position was chosen because the agreement between its $Gain()$ and $GainRatio()$ are simultaneously high, while the cut exhibits neither the highest $Gain()$ nor the highest $GainRatio()$, but it has other advantages (e.g., separability in Fig. 4).

O3: Look-ahead. Humans' insights into the consequence often influence the current decision. When interacting with parallel coordi-

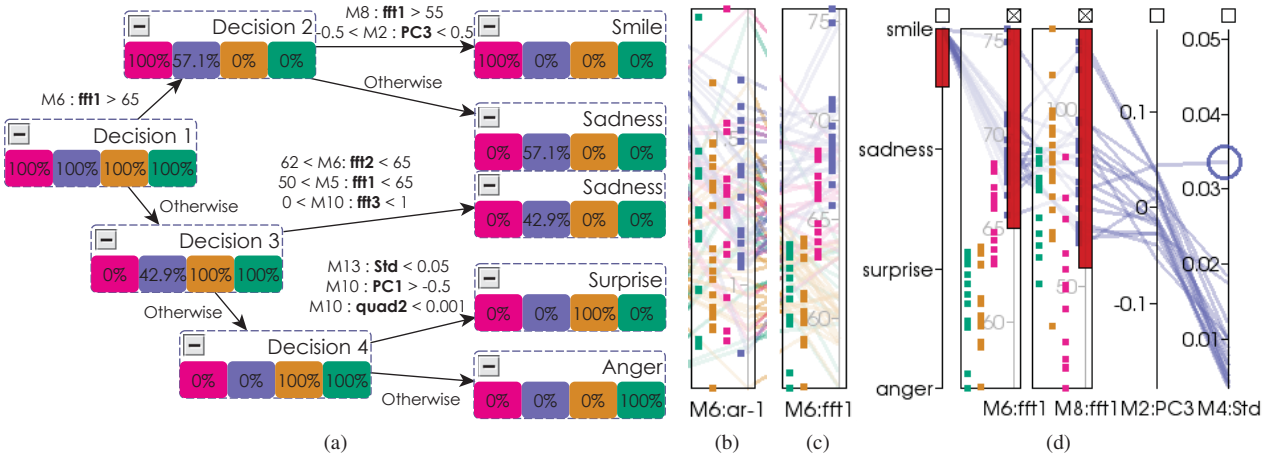


Fig. 5: (a) A decision tree constructed using the human-centric approach. Considers the top path (*smile* classification, see also Fig. 1), the model developer picked M6 (mouth width), M8 (lip curvature), and M2 (inner brow movement) (**O6**). (b) At the decision-tree root, for the same M6 measurement, M6:ar1 was found not useful, and M6:peak (see Fig. 6b) less useful because of the large overlapping pattern (**O1**, **O2**, **O6**). (c) Eventually M6:fft1 was selected. (d) At the leaf node (*smile*), M4:Std (brow horizontal movement) was not used because of an outlier (**O4**).

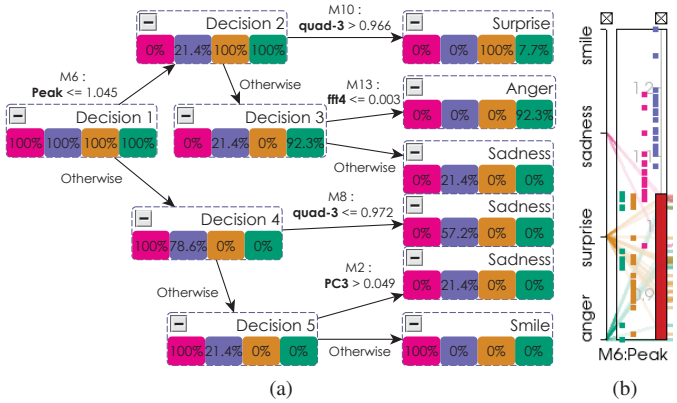


Fig. 6: (a) A decision tree created by C4.5. (b) At the root, C4.5 picked M6:peak because of its higher *GainRatio*(), but ignored the fact that *sadness* is better clustered in M6:fft1 (see Fig. 5c). It also drew the cut close to the clusters of *anger* and *surprise*. As shown in Fig. 5d, the model developer drew the cut away from the two classes (**O5**).

nates plots, model developers in **Case Studies A** and **B** often look ahead into the quality of future selection of axes and cuts. For example, in Figs. 1 and 4, a developer can easily change a selection after seeing its consequence on scatterplots for all visible axes. Thus the humans' look-ahead ability enables multi-step judgment, while the machine-learning algorithms used focus only on the current decision in each iteration.

O4: Outliers. If possible, model developers avoid axes featuring outliers, as such axes may be unreliable. In **Case Study A**, inner brow vertical (M2) and brows horizontal displacement (M4) are candidate facial features for classifying *smile*. Axis M4:Std was observed to form a stronger cluster. However, because there is an outlier (Fig. 5d, circled), M2:PC3 was used instead. Similar situations occurred in **Case Study B**. Such kind of reasoning is not available in the machine-learning algorithms. After all, automatic and accurate identification of outliers is still a challenging research topic.

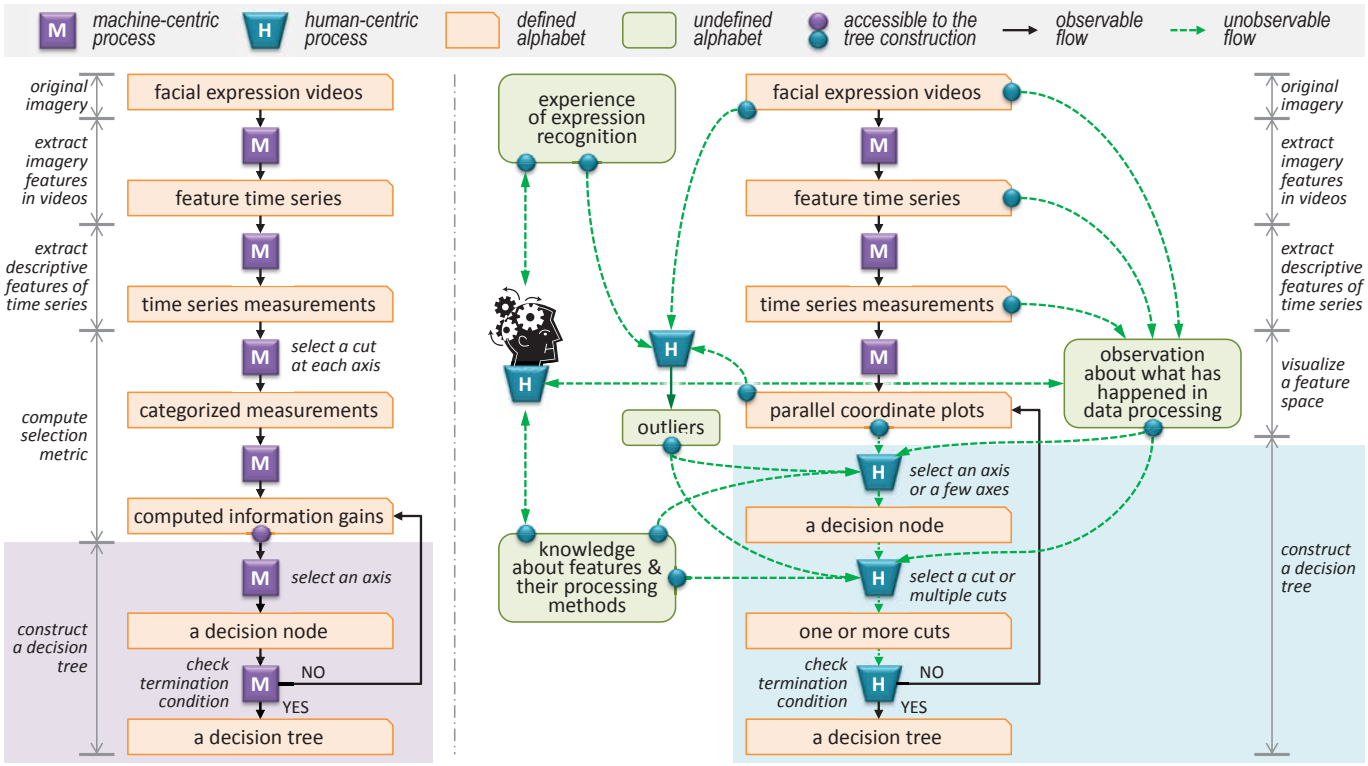
O5: Cut Positions on an Axis. When separating feature vectors at an axis, both model developers looked for a cut or cuts that would allow each class to expand beyond the current instances in the training set. For example, in **Case Study A**, the developer drew the decision value at the mid-point to separate class *smile* from both *surprise* and *anger* (Fig. 5d, M6:fft1). C4.5, however, took a cut very close to the boundary of *surprise* (Fig. 6b). This was also part of the reason why the decision tree created by C4.5 required three leaf nodes to classify *sadness* (Fig. 6a). Similar situations also occurred in **Case Study B**, where all decision boundaries were chosen to be the mid-point between data points of the respective classes to maximize margins.

O6: Human (Domain) Knowledge. In the human-centric approach, model developers incorporate their domain knowledge into the model construction process. In **Case Study A**, there were several such situations. In Fig. 5a, the choice of features in Decisions 2, 3 and 4 were not only based on data observation, but also human knowledge. For example, for Decision 2, by common sense, *smile* would associate to mouth width (M6), lip curvature (M8) and stationary brows (M2). Further, to capture such stationary brows movement, the model developer combined two rules (e.g., -0.5 < M2:PC3 < 0.5) into one. This is not possible in the three machine-learning algorithms. Another observation is that humans tend to use multiple clues together to confirm facial expressions. In Fig. 5a (Decisions 3 and 4), a *sadness* would likely have stable features in mouth width (M6), mouth height (M5), and eye size (M10). A *surprise* would likely show lots of movement in eye size (M10), but little movement in inter-brow region (M13). This high-level knowledge was captured in the human-centric decision tree (Fig. 5a), but not available to the machine-centric approach. The model created by C4.5 used only M6 and M10 to classify *surprise*, and only M6 and M8 for one of the *sadness* case (Fig. 6a). Similarly in **Case Study B**, knowing that (a) bubble charts had circles clustered around the center of the images, and (b) moment invariant was robust and invariant to scale, translation and rotation, the developer frequently picked *circular moment invariant* in the three experiments.

7 Information-Theoretic Analysis

The observations made in Section 6 can be theorized based on the theoretical proposition outlined in Section 3. Fig. 7 juxtaposes two flowcharts showing the two approaches considered in **Case Study A**. The corresponding flow charts for **Case Study B** can be easily derived by replacing the first three boxes with two boxes "visualization images" and "image feature vectors". Both charts focus on the variables that may be used directly or indirectly in the construction of a decision tree, while differentiating processing steps simply by two abstract groups: machine- and human-centric. We can consider each textbox in Fig. 7 as an alphabet – a concept described in Section 3.

From Fig. 7, we can observe that the human-centric approach for constructing a decision tree makes use of more alphabets than the machine-centric approach. For example, the alphabet of *facial expression videos*, after being transformed to *feature time series*, is no longer available to the last stage of the process on the left for constructing a decision tree. Let us numerically estimate the availability of imagery



(a) machine-centric construction

(b) human-centric construction

Fig. 7: Two flowcharts showing information flows with the two approaches constructing a decision tree in Case Study A (Section 4).

information about facial expressions. It is estimated that there are 7.4 billions of people in the world. Considered that each person may have 5 distinguishable variations for each of the four expressions, the number of valid letters in the first alphabet in both flowcharts is 148 billions. The maximal entropy is thus 37.1 bits (with an assumption of uniform distribution). The 68 videos used in **Case Study A** is just a drop in the ocean, representing 1.7×10^{-8} bits of known information.

The label of each video is of maximal entropy of 2 bits in terms of four expression classes, and for 68 videos, there are up to 136 bits in total. This information is available to both flowcharts all the time.

When the data transformation reaches *time series measurements*, there is a composite alphabet defined upon 14×23 real numbers. Let us simply assume that all letters in the alphabet *facial expression videos* are one-to-one mapped to letters in *time series measurements*, and all the information in the first alphabet are thus preserved.

The two approaches differ from this point onwards. With the machine-centric approach, the next alphabet, *categorized measurements* will likely incur information loss due to the binary grouping for each of the 14×23 measurements, though 14×23 bits can theoretically retain all 37.1 bits of entropy in the first alphabet. Optimistically, let us assume that *categorized measurements* retain 50% mutual information. In terms of known information, that is 8.5×10^{-9} bits, excluding the labeling information.

Meanwhile, with the human-centric approach, the alphabet of *parallel coordinates plots* do not have the full numerical precisions at *time series measurements*, but is capable of depicting at least 100 distinct values on each axis. Hence it will retain more mutual information than *categorized measurements*. Pessimistically, we assume 75% mutual information is retained in *parallel coordinates plots*. In addition, a model developer has a vast amount of experience of viewing and recognizing expression. For example, the model developer may know some 200 people reasonably well, and can recall their 5 variations of 4 expressions at ease. Conservatively, together with the original 68 videos, these are equivalent to 4068 videos, representing 1.0×10^{-6} bits of known information in the context of *facial expression videos*. When given an arbitrary facial image, the developer can also reconstruct an expression using imagination, e.g., at least 1 variation per expression. This ability can be translated to 29.6 billions of videos, representing 7.4 bits of known information. Such reconstructed mental videos play an important role in determining outliers.

Therefore, in terms of *facial expression videos*, a human model developer has access to 7.4 bits of known information, which is 871 millions times more than the machine-centric approach (cf. 8.5×10^{-9} bits). The above exercise of estimating information-theoretic quantities suggests that soft knowledge may come from two sources: namely *soft alphabets and letters* and *soft models*.

Soft Alphabets and Letters. This type of soft knowledge encompasses alphabets and letters that would be uncaptured or underutilized in the construction stage of a machine-centric process. For example, the videos available at the beginning of both flowcharts are underutilized letters in the machine-centric process, and the $200 \times 5 \times 4$ facial expressions that a model-developer can recall are uncaptured letters in the alphabet of *facial expression videos*. One use of these letters is in determining if an expression in a video is an outlier (Observation **O4**), which influences the selection of an axis (or axes) and the placement of a cut (or cuts) (Observation **O5**)

As observed in Section 6, a model developer may have a mental alphabet about the indicativeness of the 14 facial features in relation to each of the four expressions, e.g., at four levels, *definitely indicative*, *likely indicative*, *detectable motion*, and *no obvious motion* (**O6**). Hence, these are 14×4 alphabets, each with 2 bits of maximal entropy. To a machine, they are uncaptured alphabets, but to a human, they are usable information in selecting an axis especially among a group of axes offering a similar capacity for separating the four expressions. Similarly, the model developer may have 23 mental alphabets about the general quality of the 23 time series measurements. If each alphabet has three letters, *mostly useful*, *sometimes useful*, and *occasionally useful*, there are 1.58 bits of maximal entropy per alphabet. We collectively group such alphabets in the box *knowledge about features & their processing methods*.

In addition, the model developer may have a general overview about

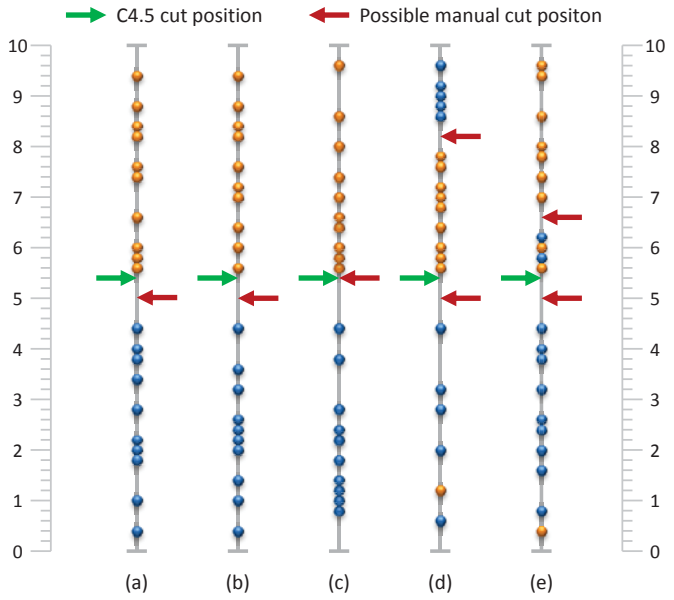


Fig. 8: Examples of five simple scenarios of selecting a cut position (or positions) to partition an axis at a decision node. (a) 2 well-separated clusters, with not well-defined distributions. (b) 2 well-separated clusters, 1 with uniform distribution and 1 with normal distribution. (c) 2 well-separated clusters, with non-linear distributions. (d) 2 intersecting clusters. (e) Another case of 2 intersecting clusters. Humans may use soft knowledge in their decisions.

available letters at each processing stage, and may make certain observation about patterns in these letters (**O1**). For example, there may be an uncaptured alphabet about the quality of the captured videos for each expression. The impression may be expressed about the number of unrealistic expressions using letters such as *almost none*, *a few*, *more than a few unrealistic*, and *a worrying number* (**O1** and **O6**). There are many such alphabets, which is difficult to define precisely. They are often dynamically created through an observation on demand, i.e., when a developer feels a need for such an observation.

Soft Models. This type of soft knowledge encompasses processes for making some decisions that would not exist in the machine-centric approach. As any process for transforming an input to an output is defined by a model, such processes are referred to as soft models. In a human-centric process of constructing a decision tree, there are many soft models, such as (i) given a facial photo (input), imagine how the person would smile (output); (ii) given a video (input), determine if it is an outlier or not (output); (iii) given a set of points on an axis (input), decide how many cuts and where they are (output); (iv) given a section of an axis with data points of different classes, predict if the entanglement can be resolved using another unused axis; and so forth.

Let us follow the discussions in Observation **O5** in Section 6. Consider an axis, as shown in Fig. 8, which has n visually distinguishable positions. There are k data points, each color-coded with its membership of a class. For the convenience of illustrative discussion, we assume that no position on the axis has more than one data point. Hence the total number of visual patterns in the input alphabet is the binomial coefficient $\binom{n}{k}$. Of course, some patterns are unlikely for this soft model. For instance, a visual pattern with alternate orange and blue dots is unlikely, since one would not choose it as an axis in the first place. Again for the convenience of illustrative discussion, we only place a cut half unit below/above a position for a data point. There are a total of $n-1$ of optional locations for a cut. Should one decides to place two cuts, there are $(n-1)(n-2)$ options. The maximal entropy for the output alphabet with 1 or 2 cut positions is $2\log_2(n-1)$. For a relatively low resolution of parallel coordinates plot with 101 distinguishable positions for data points, the maximal entropy is of 13.3 bits. Before the model developer makes a decision about cut points, there are 13.3 bits of uncertainty. Once a decision is made, the soft model generates what equivalent to 13.3 bits of known information.

In addition, there are other soft models used in the human-centric approach, such as determining what is a cluster (**O1**), multi-factor decision (**O2**), and lookahead prediction (**O3**). Each of these generates what equivalent to many bits of known information.

A table in the supplementary materials summarizes the estimated amount of soft knowledge available in **Case Studies A** and **B**. It is necessary to note that the availability is not the same as utilization. There is a cost for using such soft knowledge, as theorized in [11]. In general, a model developer would limit the use of soft knowledge to a small number of situations. For example, among the five cases in Fig. 8, the developer may not call a soft model for checking outliers at all in the cases (a-c). In the cases of (d) and (e), one may assume that the orange dot at the bottom end of the axis is to be an outlier without viewing the corresponding video. Meanwhile one may decide to investigate the videos for the 2 orange and 2 blue dots in the middle of (e) in detail. If the two orange dots are judged as outliers, a cut point may be placed near 6.6. If the two blue dots are judged as outliers, a cut point may be placed near 5. If none of these is an outlier, two cut points may be used to segment the overlapping range out, and that can subsequently be separated by another axis in the next iteration.

8 Conclusions

In this paper, we presented an in-depth analysis of two visual analytics case studies, where both machine- and human-centric approaches were used to construct decision-tree models for classification tasks. One case study is a new application for automated classification of the type of visualization images, while the other was previously reported in [47] for facial expression classification. In both case studies, the human-centric approach produced better decision trees than the machine-centric approach, which is somehow surprising as there is wide optimism about machine learning. In our in-depth analysis, we first collected various empirical evidence that suggest a human model developer may have done differently from the machine-learning algorithm. We then theorize the observations using information theory. In particular, we quantitatively estimated Shannon entropy of various alphabets featured in the two pipelines corresponding to machine- and human-centric approaches. The estimation shows that there are overwhelming amount of information available to the human-centric approach. Some information is in the form of *soft alphabets* encompassing human knowledge that are not captured in the data. Other results from *soft models*. Although there is a huge cost for accessing additional information, this cost is often significantly reduced by the presence of soft models that generate instances of an alphabet (e.g., cuts, outliers, etc.) on demand. Our in-depth analysis has provided a theoretical justification about the merits of visual analytics exhibited in these two case studies.

However, the analysis and theorization does not in any way suggest that one should cast aside machine learning. On the contrary, we would like to make the following arguments. First, it is necessary to analyze the information flow as exemplified in this work before determining the roles of machine and human in the process of developing a model. Second, it is helpful to increase the number of digitally stored alphabets and the number of instances in each alphabet. This can reduce the dependency on human model developers' ad hoc alphabets, while improving the accuracy of the probability distribution within each alphabet, which is critical to most machine-learning processes. Third, it is helpful to consider each classification model not as an isolated model, as it often requires assistance of other models. A high quality model will likely be built in conjunction with other high quality models. Last but not least, one should never underestimate human model developers' ability to supply new soft alphabets and soft models that are not in the system. The goal is to enable human model developers rather than casting them aside.

References

[1] B. Alsallakh, A. Hanbury, H. Hauser, S. Miksch, and A. Rauber. Visual methods for analyzing probabilistic classification data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1703–1712, 2014.

[2] M. F. Amasyali and O. K. Ersoy. Comparison of single and ensemble classifiers in terms of accuracy and execution time. In *Proc. Innovation Intelligent System Applications*, 2011.

[3] R. K. Anderson. *Visual Data Mining: The VisMiner Approach*. Wiley-Blackwell, 2012.

[4] M. Ankerst, C. Elsen, M. Ester, and H.-P. Kriegel. Visual classification: An interactive approach to decision tree construction. In *Proc. 5th ACM SIGKDD*, pages 392–396, 1999.

[5] M. Babaee, S. Tsoukalas, G. Rigoll, and M. Datcu. Visualization-based active learning for the annotation of SAR images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(10):4687–4698, 2015.

[6] H. Bay, T. Tuytelaars, and L. Van Gool. Speeded up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[8] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, 1984.

[9] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for health care: Predicting pneumonia risk and hospital 30-day readmission. In *Proc. 21th ACM SIGKDD*, pages 1721–1730, 2015.

[10] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proc. 23rd International Conference on Machine Learning*, pages 161–168, 2006.

[11] M. Chen and A. Golan. What may visualization processes optimize? *IEEE Transactions on Visualization and Computer Graphics*, online preprint, 2015.

[12] M. Chen and H. Jänicke. An information-theoretic framework for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1206–1215, 2010.

[13] S. Cheng and K. Mueller. The data context map: Fusing data and attributes into a unified display. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):121–130, 2016.

[14] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.

[15] R. J. Crouser, B. Hescott, and R. Chang. Toward complexity measures for systems involving human computation. *Human Computation*, 1(1):4565, 2014.

[16] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc. Workshop on Statistical Learning in Computer Vision*, pages 1–22, 2004.

[17] P. Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, Oct. 2012.

[18] H. L. Dreyfus and S. E. Dreyfus. What artificial experts can and cannot do. *AI & Society*, 6(1):18–26, 1992.

[19] M. Fahle and T. Poggio. Visual hyperacuity: Spatiotemporal interpolation in human vision. *Proceedings of the Royal Society of London B: Biological Sciences*, 213(1193):451–477, 1981.

[20] H. Fang, N. M. Parthaláin, A. J. Aubrey, G. K. L. Tam, R. Borgo, P. L. Rosin, P. W. Grant, D. Marshall, and M. Chen. Facial expression recognition in dynamic sequences: An integrated approach. *Pattern Recognition*, 47(3):1271–1281, 2014.

[21] H. Fang, G. K. L. Tam, R. Borgo, A. J. Aubrey, P. W. Grant, P. L. Rosin, C. Wallraven, D. W. Cunningham, D. Marshall, and M. Chen. Visualizing natural image statistics. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1228–1241, 2013.

[22] S. García and F. Herrera. An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

[23] T. Hancock, T. Jiang, M. Li, and J. Tromp. Lower bounds on learning decision lists and trees. *Information and Computation*, 126(2):114–122, 1996.

[24] Z. S. Harris. Distributional structure. *Word*, 10(23):146162, 1954.

[25] B. Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, and G. Heidemann. Inter-active learning of ad-hoc classifiers for video visual analytics. In *Proc. IEEE Conference on Visual Analytics Science and Technology*, pages 23–32, 2012.

[26] L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1):15–17, 1976.

[27] L. Itti and P. F. Baldi. Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems, Vol. 19 (NIPS*2005)*, pages 547–554, 2006.

[28] A. Kapoor, B. Lee, D. Tan, and E. Horvitz. Interactive optimization for steering machine classification. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pages 1343–1352, 2010.

[29] A. Kapoor, B. Lee, D. Tan, and E. Horvitz. Performance and preferences: Interactive refinement of machine learning procedures. In *Proc. AAAI Conference on Artificial Intelligence*, 2012.

[30] H. Kim, J. Choo, H. Park, and A. Endert. Interaxis: Steering scatterplot axes via observation-level interaction. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):131–140, Jan 2016.

[31] S. B. Kotsiantis. Supervised machine learning: a review of classification techniques. *Informatica*, 31(3):249–268, 2007.

[32] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Dept. Computer Science, University of Toronto, 2009.

[33] P. A. Legg, D. H. S. Chung, M. L. Parry, R. Bown, M. W. Jones, I. W. Griffiths, and M. Chen. Transformation of an uncertain video search pipeline to a sketch-based visual analytics loop. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2109–2118, 2013.

[34] Z. Liao, Y. Yu, and B. Chen. Anomaly detection in gps data based on visual analytics. In *Proc. IEEE Symposium on Visual Analytics Science and Technology*, pages 51–58, 2010.

[35] T. Löwe, E. C. Förster, G. Albuquerque, J. P. Kreiss, and M. Magnor. Visual analytics for development and evaluation of order selection criteria for autoregressive processes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):151–159, 2016.

[36] T. May, A. Bannach, J. Davey, T. Ruppert, and J. Kohlhammer. Guiding feature subset selection with an interactive visualization. In *Proc. IEEE Conference on Visual Analytics Science and Technology*, pages 111–120, 2011.

[37] M. Migut and M. Worring. Visual exploration of classification models for risk assessment. In *Proc. IEEE Symposium on Visual Analytics Science and Technology*, pages 11–18, 2010.

[38] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[39] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100), cucs-006-96. Technical report, Columbia University, 1996.

[40] S. J. D. Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, 2012.

[41] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[42] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, 1993.

[43] M. Röhlig, M. Luboschik, F. Krüger, T. Kirste, H. Schumann, M. Bögl, B. Alsallakh, and S. Miksch. Supporting activity recognition by visual analytics. In *Proc. IEEE Conference on Visual Analytics Science and Technology*, pages 41–48, 2015.

[44] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, D. Weiskopf, S. C. North, and D. A. Keim. Human-centered machine learning through interactive visualization: Review and open challenges. In *Proc. 24th European Symposium on Artificial Neural Networks*, 2016.

[45] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

[46] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45(4):427 – 437, 2009.

[47] G. K. L. Tam, H. Fang, A. J. Aubrey, P. W. Grant, P. L. Rosin, D. Marshall, and M. Chen. Visualization of time-series data in parameter space for understanding facial dynamics. *Computer Graphics Forum*, 30(3):901–910, 2011.

[48] C.-H. Teh and R. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):496–513, 1988.

[49] S. van den Elzen and J. van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *Proc. IEEE Conference on Visual Analytics Science and Technology*, pages 151–160, 2011.

[50] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. 32nd International Conference on Machine Learning*, pages 2048–2057, 2015.

[51] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. In *Proc. Deep Learning Workshop, 32nd International Conference on Machine Learning*, 2015.