An Approach to Supporting Incremental Visual Data Classification

Jose Gustavo S. Paiva, William Robson Schwartz, Helio Pedrini, and Rosane Minghim

Abstract—Automatic data classification is a computationally intensive task that presents variable precision and is considerably sensitive to the classifier configuration and to data representation, particularly for evolving data sets. Some of these issues can best be handled by methods that support users' control over the classification steps. In this paper, we propose a visual data classification methodology that supports users in tasks related to categorization such as training set selection; model creation, application and verification; and classifier tuning. The approach is then well suited for incremental classification, present in many applications with evolving data sets. Data set visualization is accomplished by means of point placement strategies, and we exemplify the method through multidimensional projections and Neighbor Joining trees. The same methodology can be employed by a user who wishes to create his or her own ground truth (or perspective) from a previously unlabeled data set. We validate the methodology through its application to categorization scenarios of image and text data sets, involving the creation, application, verification, and adjustment of classification models.

Index Terms—Visual image classification, multidimensional point placement, information visualization

1 Introduction

Data classification is involved in a large variety of data intensive tasks and applications. However, no classification technique produces good results in all scenarios, and they need to be adapted to the data at hand. Final classification results strongly depend on several factors, such as quality of the feature space and employed similarity measure. Adequacy of the training set is also crucial [14].

Users play an important role in building, applying and adjusting classifiers, since their knowledge of the problem allows adequate instance set selection as well as faster understanding of the reasons for poor classification results. This is particularly the case for exploratory analysis, where samples sets are not properly assigned or even possible to attain. Active learning (AL) [20] techniques create a process in which a classifier is interactively trained with user's annotations on informative samples, allowing the classifier to choose the data from which it wants to learn. The idea is that a classifier trained on a small set of well-chosen examples can perform as well as a classifier trained on a larger number of randomly chosen examples, requiring much less computational effort [45]. These techniques thus provide means to the user to insert his or her knowledge about a specific scenario in the refinement of the classification model, maximizing its generalization capabilities.

User interference in the classification process may be potentially more effective if the data sets are presented in a meaningful and effective manner, so he or she can easily understand its structure, the relationship amongst instances, and detect important data specificities that justify classifiers behavior. Thus, a complete and consistent visual analysis approach to supporting classification, coupled with appropriate visualization techniques, can have considerable impact in the outcome of a large variety of data analysis challenges.

Point-based visualizations can be successfully used to give such support, since they strive to display data points that are highly related in the same region or sector of the layout. Among them, similarity trees [7] with an appropriate radial layout [2] have been shown to lead to adequate displays for classification related tasks [29]. Multidimensional projections, which have gained considerable attention lately, also provide suitable layouts.

In this paper we propose a *Visual Classification Methodology* (VCM) that integrates point-based visualization techniques into classification pipelines to support control over the whole classification process. Our hypothesis is that point-based visualization techniques allow insights into the data set structure that improve the comprehension of the classifiers behavior, supporting an interactive and iterative user insertion in the classification process for convergence to adequate results. Any supervised categorization method can be supported by our approach, and the process is incremental, allowing progressive adjustments to the model, a necessary requirement for current dynamic data sets.

Since updating of the classifier is fundamental to handle such evolving data sets, a model that is incremental would also improve time and memory requirements. We exemplify our approach and the system implemented to realize it through the use of an incremental classification technique [49].

- *J.G.S. Paiva is with the Faculty of Computer Science, Federal University of Uberlandia-UFU, Uberlandia, Minas Gerais, Brazil. E-mail: gustavo@ufu.br.*
- *W.R. Schwartz is with the Department of Computer Science, Federal University of Minas Gerais-UFMG, Belo Horizonte, Minas Gerais, Brazil. E-mail: william@dcc.ufmg.br.*
- *H. Pedrini is with the Institute of Computing, University of Campinas-UNICAMP, Campinas, Sao Paulo, Brazil. E-mail: helio@ic.unicamp.br.*
- *R. Minghim is with the Institute of Mathematics and Computer Science, University of Sao Paulo-USP, Sao Carlos, Sao Paulo, Brazil. E-mail: rminghim@icmc.usp.br.*

Manuscript received 22 Aug. 2013; revised 16 May 2014; accepted 8 June 2014. Date of publication 18 June 2014; date of current version 26 Nov. 2014. Recommended for acceptance by H. Qu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2014.2331979

In this context, the main contributions of this work are:

- A visual classification methodology for incremental building and adjustment of classification models as well as a system that instantiates the methodology.
- A coupled similarity based strategy for analysis of classification results and support to feedback into the classification process.
- The validation of the methodology through a series of case studies.

We evaluate the approach in various scenarios, including that of strong structural changes in the collections, such as when new classes appear.

The following sections describe related work, our approach to interactive visual classification and underlying methods, the results of several classification scenarios for text and image data sets, and the illustration of both trees and projections as supporting visual tools.

2 Background and Related Concepts

This section presents current research on data classification, active learning and visual classification, as well as the classification and visualization approaches used in our experiments. It is important to highlight, however, that our proposed methodology supports the use of many other classification methods or visualization strategies.

2.1 Data Classification and Active Learning

Several authors [21], [35], [52] advocate the importance of user's insertion in the image retrieval and classification processes, combining human knowledge with computational capabilities to improve the confidence and comprehensibility of the model created. This insertion is necessary because of many factors. First, many types of information, such as images and videos, reside in a continuous representation space [53], in which semantic concepts are best described in discriminative subspaces. Thus, only a small subset of this space is not enough for describing all concepts. Second, users tend to employ high-level concepts to interpret images and measure their similarity, while mostly low-level features are automatically extracted, producing a semantic gap [25]. Additionally, different users at different times may have distinct interpretations or intended usages for the same images, which restricts the reach of an offline and user-independent learning. Finally, most classification algorithms aim at fully automatic procedures, preventing users from understanding the decisions made by the classifier, and from inserting their domain knowledge into the classification [4].

Relevance feedback (RF) [5] approaches can be used to provide user insertion. Through them, classification models are adjusted using information possibly collected via user interaction. Users may indicate instances of interest or instances that could be excluded, and these interactions help adjust the classifier to user preferences. Following this idea, active learning [20], in which the classifier is interactively trained with user's annotations on informative samples is also employed. The idea is that a classifier trained on a small set of well-chosen examples can perform as well as a classifier trained on a larger number of randomly chosen examples, requiring much less computational effort [45], and reducing human-labeling burden. These systems show the user a set of instances from which the classification result is most uncertain and these instances will be used to reinforce the model knowledge and maximize its generalization capabilities from an accurate labeling performed by the user. The use of support vector machines (SVM) associated with AL is a popular approach [43]. Conventional SVM, however, does not take into account the redundancy among training instances, allowing similar or even identical selections, leading to suboptimal solutions [17].

Incremental learning schemes are especially suitable for real applications since not all information contained in training sets is useful to the classifier and because human learning is a gradual process. Some learning techniques employ generative probabilistic models that learn incrementally by using a Bayesian strategy [12], or incremental SVMs [24]. A framework with an incremental multi-class SVM classifier is also proposed for large scale unbalanced image annotation [44], with good performance reached with just a small portion of the collections annotated. Other AL techniques can be found in [38], [50].

2.2 Visual Data Classification

Classification systems that employ AL may incorporate visual interactive tools that allow user insertion in the process. However, many approaches limit user actions to answering questions about the relationship amongst selected instances [20], or to selecting, from a list, relevant or irrelevant samples. In order to ensure a complete experience to user's classification activities, it is necessary to display the data in a meaningful manner, so they can understand the structure and specificities of the data set. Rodden et al. [34] demonstrate the potential of organizing images by their similarity using layouts created by multidimensional scaling (MDS) techniques, adapted to fit the images in square grids and thus avoid overlapping. Nguyen and Worring [28] compare traditional sequential visualization with a similarity-based approach that associates stochastic neighbor embedding (SNE), locally linear embedding (LLE) and ISOMAP, in a scenario of image annotation. The approach reduces the total annotation effort significantly, requiring 16 times lower effort depending on the separation of the different categories. These alternatives provide good evidence for the utility of similarity based displays, but the support given to the process is still limited.

Some systems aim at supporting understanding of several mining procedures [3], [18], [23]. The iVisClassifier [4], that employs linear discriminant analysis (LDA) to perform a dimensionality reduction, focuses on group discriminability in order to ease the labeling of new instances. The system permits a recomputation of LDA considering new instances labeled by the user, representing the incremental insertion of his or her knowledge to the process, but it does not provide further interference with the classification process. Users also have to label each instance, another limitation of the approach.

In [23], the authors highlight the difficulty, in the image and video search scenarios, of adequately annotating instances, as well as the lack of suitable training data. They have developed a visual analytics system that gives visual

feedback to users, along with a normalized maneuver visualizations to explore the video data. Users can also choose to accept or reject particular results to train the system, enhancing the similarity visualization accordingly and incorporating human knowledge into the model. The result is a solution that begins to reduce the ambiguity that user sketches could have on an untrained system, creating a more powerful analytical system for the end-user.

Another example of a visual AL application [26] shows the decision boundary of the classifier and its correspondent performance curve. It allows changes in the classification model by moving points on this performance curve, modifying the suitable tradeoff for a specific application, or by assigning a different label to an instance and using it to retrain the classifier. A similar system [15] also shows the decision boundary of the classifier, using a scatterplot matrix, with its axes representing the confidence value for each document and the diversity of the documents closest to the decision boundary. It also presents another view that uses a least squares projection (LSP) [31] to show the produced clusters. This system is based on the SVM classifier, and the user can modify or assign labels to the instances by selecting them in one of the views. These instances will be used to retrain the classifier. These systems are applicable only to binary classification problems that encompass decision boundaries, which may not be trivial to construct and interact with in multi-class scenarios.

Interactive construction of decision tree classifiers has also been proposed [8], [46], which provide means to optimize and prune a decision tree, and analyze it together with an underlying structure of the data with linked views and interaction techniques. These are specific to decision trees and employ attribute based visualizations, which do not scale for high dimensional data.

Finally, an *inter-active* learning system [16] provides feedback on classification quality to users by means of a set of integrated cascaded scatter plots of the instances class distribution, in each stage of the classifier. Annotated instances are also organized by their similarity, using a t-distributed stochastic neighbor embedding (tSNE) [47]. The system provides a set of interaction tools that allow the annotation of selected instances, and the assessment of the quality of the classifier, discovering regions with wrong class assignment. In this paper, we propose a similar classification methodology, but visually exploring the assessment and comprehension, by the user, of the data structure, and its influence on the classifier behavior, with feedback into the classification process.

Previous work mentioned above show that classification tasks are properly supported by visualization techniques that highlight the relationship amongst instances. In our work, we show that multidimensional projections and similarity trees consistently support various stages of and incremental data classification for general classifiers. We choose similarity trees and projections for this support, the first offering additional levels of detail in terms of degrees of similarity between instances compared to the latter.

2.3 Related Concepts

In this section we give further details of the particular incremental classification model employed in the experiments for this paper, and also on point layout for data set visualization, the latter being the visual basis for our interaction with the classification processes.

2.3.1 Locally Weighted Projection Regression (LWPR)

The Locally Weighted Projection Regression [48], [49] algorithm estimates a nonlinear function approximation in high dimensional space, even in the presence of redundant and irrelevant input dimensions. It uses a set of locally linear models spanned by a small number of univariate regressions in specific directions in the original input space. An online weighted version [49] of Partial Least Squares (PLS) algorithm [51] is used to perform dimensionality reduction on the specific directions. LWPR has been widely employed in prediction tasks in several areas, such as Medicine [13], Computer Aided Design [27], Robotics [9] and Civil Engineering [1]. This section briefly presents a mathematical description of the technique.

An LWPR regression model is constructed through a set of training instances represented as vectors $\mathbf{x}_i$ and correspondent responses y_i, iteratively presented as input-output tuples $(\mathbf{x}_i, y_i)$. The LWPR prediction will be the weighted sum of the prediction of each of k locally linear model, according to Equation (1):

$$\widehat{y} = \frac{\sum_1^k w_k \widehat{y}_k}{\sum_1^k w_k} \tag{1}$$

The weights w_k of each locally linear model define the validity area of these models, also called *Receptive Fields*, and are usually modeled by a Gaussian kernel, according to Equation (2), in which $\mathbf{D}_k$ represents a distance metric and $\mathbf{c}_k$ represents the center of the kernel. The distance metric determines the size and shape of the validity area for each model. A weight value $w_{i,k}$ is computed for each input instance i. This weight varies according to the distance to the center of the kernel of each model k, which makes the learning process of a model localized and independent of the other models. If the activation value of all the existing models is lower than a threshold, a new model is created, granting that the number of models is dynamically updated during the process:

$$w_{i,k} = \exp(-0.5(\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{D}_k (\mathbf{x}_i - \mathbf{c}_k)). \tag{2}$$

The online weighted version of PLS is employed in the learning of the locally linear models, so that, for each model, the dimensions of the input instance $\mathbf{x}_i$ are sequentially regressed along selected projections u_r, chosen by the technique in input space, yielding a set of r latent variables. These directions are chosen according to the correlation of the input data with the output data (class information). The regression on a locally model will be composed by the linear combination of the latent variables for this model. More details regarding the online PLS algorithm can be consulted in [49].

The distance metric $\mathbf{D}_k$ of each receptive field is individually updated, using an incremental gradient descent based on stochastic leave-one-out cross-validation criterion [48]. Algorithm 1 illustrates the incremental learning process of an LWPR model. In the algorithm, w_{gen} represents a

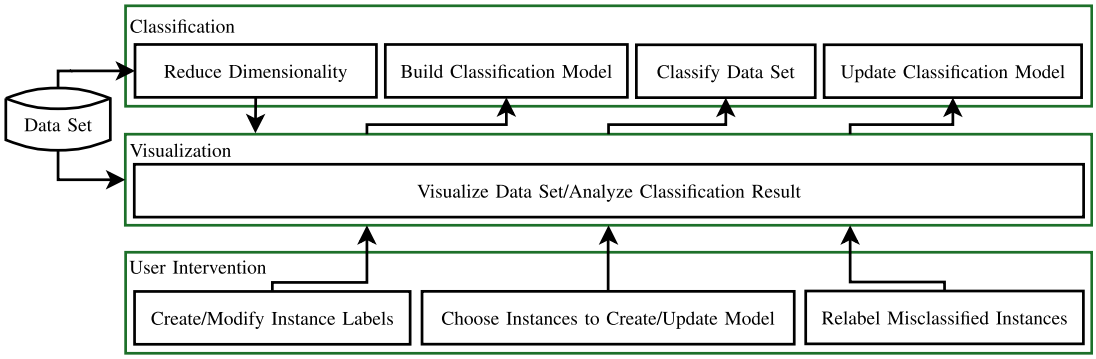


Fig. 1. Visual classification methodology diagram, showing three perspectives: CLASSIFICATION, that concentrates the automatic classification flow, VISUALIZATION, that provides the visual support to comprehension of the data set and classification processes, and USER INTERVENTION, that provides the possible manners for the user to influence the process by interacting with visualizations.

threshold for creation of new local models, and r_k is the number of latent variables for each of the k local models.

Algorithm 1. *Locally Weighted Projection Regression*. Adapted from [48]

```
Initialize LWPR model with no local models (LM = 0)
for all training instance (x_i, y_i) do
  for k = 1 : LM do
    Calculate activation value (Equation (2))
    Update local PLS model and D_k
  end for
  if no linear model was activated by more than w_gen then
    Create a new local model with r_k = 2, c_k = X, D_k = default value
  end if
end for
```

2.3.2 Point-Placement Strategies

Point-based visualization strategies are used to map data instances into a visual display, offering a solid first step to explore data sets. These strategies rely on relationship among individuals calculated using all the available attributes. Multidimensional projections can be used to generate these mappings, and most of them rely on the final result placing highly related individuals in the same region in visual space. Interpretation of the layout is accomplished by locating groups of interest and focusing on such groups and their subgroups.

Several multidimensional projection techniques are available, most of them based on dimension reduction techniques. A largely used approach is Principal Component Analysis (PCA) [19], that employs linear combinations of the data attributes (dimensions) with a high covariance degree, producing directions with less dependence. Multidimensional scaling [6] comprises a class of techniques that can be used to perform projections. The simplest MDS approach, Force Directed Placement (FDP) [10], is based on a spring system concept, where the multidimensional instances are modeled as objects linked by springs such that the repulsion and attraction forces between the objects are proportional to the multidimensional distances. The projection is attained when the spring system reaches an equilibrium state.

Regardless of their advantages, multidimensional projections present drawbacks that may impair the comprehension of the data set by users, such as the difficulty in maintaining locally the same levels of precision made globally, and the high degree of cluttering produced. As an alternative, the data set can be organized as a similarity tree. Neighbor Joining (NJ), a phylogeny reconstruction algorithm, has been adapted to construct similarity trees [7], [29], in which leaves represent instances, internal nodes represent hypothetical ancestors, and edge lengths indicate the distance among instances. By positioning objects on branches, similarity is organized into levels, an intuitive way of interpreting degrees of similarity. Global analysis is not impaired, and local analysis is as precise as global.

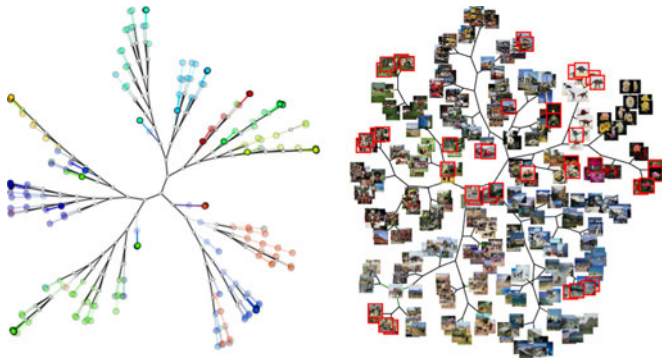
3 Visual Incremental Classification

We additionally hypothesize that interaction in every step of the classification process can be benefited by users recognizing groups of highly related individuals. Fig. 1 illustrates the functional structure of the VCM, totally supported by a similarity-based form of visualization, that is, a point placement strategy that reflects groups of similar individuals.

This section describes the steps of the VCM methodology. Although VCM works with any classification methods based on training from samples, it benefits most when an incremental method is employed. An incremental classification method makes it faster to rebuild and apply the model. We have implemented various methods, but present the approach using LWPR, described in Section 2.3.1.

3.1 Creating and Applying the Model

The instances that compose the training for the classification model are informed by the user. They can be selected using the visualization layout, whose structure and point organization is able to guide the user towards a relevant selection. Figs. 2a and 2b show two views of a Neighbor Joining tree for a set of 300 images, from which the user selected 43 images as a training set. In Fig. 2b, a fast simplified force-based layout [42] was applied in order to minimize cluttering and allow inspection of the branch organization. In this case, the images from the end of branches were chosen together with images from the core of the tree. The



(a) Selected instances. (b) Selected images.

Fig. 2. NJ tree for a 300 image collection, with 43 selected instances, represented by circles (2a) and by images (2b).

confidence of the classification is higher in instances located at the end of the branches and lower in instances located at the top of the branches. In the system it is possible to assign and change labels of selected instance.

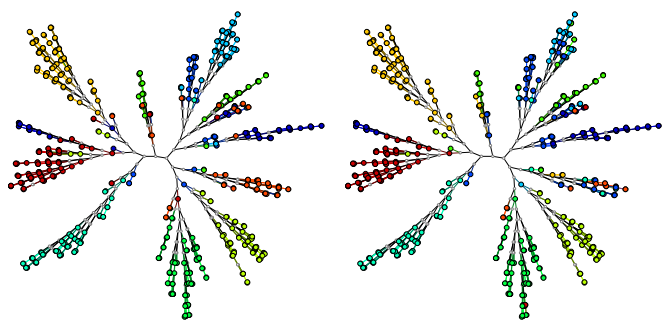
The created classification model can be employed in the classification of any collection bearing the same configuration for the feature space. Fig. 3a shows an NJ tree with the ground truth of another image collection with 700 images. Fig. 3b shows the visual classification result for this collection, using the LWPR model created from the samples shown in Fig. 2. Numerical results are shown in Table 1.

3.2 Updating the Model

Model updates can also be performed by selecting additional instances from a visualization layout. In LWPR, model learning is performed incrementally, thus the selected instances will update each PLS linear model locally, generating new ones if necessary, according to the algorithm formulation presented in Section 2.3.1. The updated model will accumulate information from instances used both in creation and update procedure, so no actual instances need to be stored.

Several model updating strategies can be adopted. In this work, we use a set of misclassified instances to add information about the classes for which the model is deficient. The similarity layout may also serve for this purpose, helping users identify the reasons for failure by looking at the images deemed similar to the misclassified ones.

When a non-incremental model is used for classification (the system allows, for instance, the use of non-incremental PLS and SVM), the update module in fact rebuilds that model using original plus added training instances.



(a) Ground truth. (b) Classification result.

Fig. 3. Classification result for a collection with 700 images.

TABLE 1
Numerical Results of COREL-700 Classification

Results	Values
Matching Instances	599 (85.6%)
Non-matching Instances	101 (14.4%)
Accuracy	97.43%
Precision	87.6%
Recall	85.57%

Table 2 presents the model creation and update times, comparing an incremental approach (LWPR), and two non-incremental approaches (PLS and SVM), for a collection with 21,643 instances categorized in eight classes, using an initial sample set of 1,200 instances, and using an update sample set of 24 instances, three of each class. One can notice that the creation of a LWPR model is slower than the creation of the PLS and SVM models, but the LWPR update process is considerably faster. Moreover, the time required to update the PLS and SVM models are similar to the time required to create these models, since these models are actually rebuilt.

3.3 Rebuilding the Model with New Classes

In some classification scenarios, it is possible that some drastic changes in the class distribution occur and new classes appear. In this case, even for LWPR, one or more models have to be created or rebuilt. To avoid re-entering previous training instances, we opt to store the training sets from previous model updates. This information is then used in an update process involving instances of n new classes.

Using the One-Against-All approach, instances of the current update that belong to the n new classes are employed in the update of the existing models. Then, n new models are created, and the previously stored instances, together with the new ones, are used to train them, resulting in $C + n$ models. The model created through the Multiclass-Matrix approach will require the addition of n new responses. Thus, a new model is created with $C + n$ responses, and the previously stored instances, together with the new ones, are used to train it. In both approaches, the instances used in the current update are also stored, together with the previous ones, so they can be used in future updates that involve new classes. For other classifiers we also re-create the model from stored training instances.

3.4 The Visual Classification System (VCS)

A Visual Classification System, made available at http://vicg.icmc.usp.br/infovis2/Tools, has been implemented aiming at creating the environment presented in Fig. 1, with the entire classification process supported by visualization techniques. Its functionalities and their relation with the VCM are described as follows:

TABLE 2
Time Comparison (in Seconds) to Create and Update Incremental and Non-Incremental Models

	LWPR		PLS		SVM
	O-A-A	*Multiclass*	*O-A-A*	*Multiclass*	
Model Creation	36.83	32.07	9.09	29.13	2.66
Model Update	0.75	0.65	8.30	19.17	2.91

Visualization and creation of the training set. The VISUALIZATION module employs a visualization technique to construct a layout that allows the user to see the structure of the collection, as well as to detect trends and patterns that may provide a better comprehension of the classification process. This layout is presented in the main screen of the system, from which the user selects instances by clicking on them, or by selecting regions of the layout. It is also possible to visualize the content of each instance. All the interaction tools are implemented in the USER INTERVENTION module. The selected instances will compose the training set used to build the classification model. Eventually, before generating the layout, a dimensionality reduction procedure can be employed in the collection to highlight in its structure some specific perspective. If the collection is not previously labeled or if the user is not satisfied with the current label scheme, he or she can create or modify the label for any instance on the layout being inspected.

Building and application of the classification model. The CLASSIFICATION module includes all the automatic classification steps, including model building using the set of selected training instances, and its subsequent application to a test collection. This model can also be saved for use in future applications. Various classifiers are included in the system, such as SVM, PLS and LWPR.

Visualization and analysis of the classification results. The VISUALIZATION module presents the test collection classification results, using the same visualization strategy employed to create the training set. The USER INTERVENTION module contains a set of tools to analyze the classification results. In situations in which a ground truth exists for a collection, it is possible to perform this analysis with classical evaluation measures, such as accuracy, precision and recall, as well as to check the number of misclassified instances and the associated confusion matrix. Additionally, the system provides a tool named *Class Matching*, that uses the layout to visualize, in contrasting color, the individuals that were misclassified, to understand its relationships to the neighboring data points.

Update of the classification model. By analyzing the layout of the classification results, in the VISUALIZATION module the user can select instances to update the classification model, also by clicking on them, or by selecting regions of the layout. The layout acts here as a guide to ease this selection, and enables several update strategies. He or she can also modify the instances labels, using the USER INTERACTION module tools. The selected instances will be used to update the model, that will again be applied to the test collection by the CLASSIFICATION module. In this sense, the user participates in an iterative process, in which as many model updates as necessary can be performed, seeking its adaptation to a particular classification scenario, and converging to the desired results.

4 Experimental Results

This section presents the results of several case studies representing classification scenarios undertaken with VCM, by means of the system built to realize it, VCS. The goal is to offer evidence of a visual framework to provide the insertion of the user into the classification process of a potentially evolving data set.

4.1 Data Sets and Test Setup

One textual and two image data sets were employed in the evaluation tests. The ALL data set, made available at vicg.icmc.usp.br/infovis2/DataSets, contains 2,814 abstracts of scientific papers in nine areas of knowledge, collected from various sources, with considerable part of common content across labels. From the text set, a feature space was created by removing stopwords and employing stemming [32]. The coordinate of any particular point was determined by the *term-frequency-inverse-document-frequency* (TD-IDF) count [39], which has been employed successfully in text visualization and mining applications, resulting in instances with 5,163 dimensions. The nine labels of the data set were assigned manually based on the perceived main topic of the scientific paper. The number of papers across labels is largely unbalanced.

The ETHZ image collection represents a subset of the ETHZ data set [11], [36], which provides photographs of different people captured in uncontrolled conditions, with a range of appearances. This collection is composed of 2,019 images, divided into 28 labels forming unbalanced groups. Each image is represented by a vector of 3,963 descriptors, combining Gabor filters, Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and mean intensity, the same setup used in [37] for face recognition.

The Free-Photo image collection is a set of 3,462 images divided into nine unbalanced classes made freely available at www.freefoto.com. Each image is described by 128 BIC (Border/Interior Pixel Classification) [33] features.

The experiments presented here were performed on an Intel Core i7 processor with 3.40 GHz and 16 GB RAM, using an LWPR library [22] wrapped in a JAVA package, that contains all the methods for model creation and update, as well as the classification methods. The dimensionality of ALL and ETHZ collections were reduced by a PLS dimensionality reduction procedure [30], in its *MulticlassMatrix* modality, and training sets containing 647 instances for ALL collection, and 400 instances for ETHZ collection. The ALL and ETHZ instances were reduced to 63 and 48 dimensions, respectively. This dimensionality reduction was performed aiming at highlighting the separability amongst classes on these collections, easing the selection of representatives by the user.

To evaluate the classification process, we measured the number and percentage of *matching instances*, representing instances correctly classified, and *non-matching instances*, representing misclassified instances. We also used the following measures, for each class: *accuracy*, which measures the proportion of correctly classified instances, amongst all instances; *precision*, which measures the proportion of correctly classified instances, amongst all instances categorized into the same class; and *recall*, which measures the proportion of correctly classified instances, amongst all instances that really belong to this class. We employed the average of these values to evaluate the classification process.

4.2 Impact of Instance Positioning

This experiment aims at evaluating how the instance position in similarity trees may impact the LWPR model

TABLE 3
Results of Classification Using Three Types of Training Set

	External Instances	Internal Instances	Combined Instances
ETHZ-Reduced			
Matching Instances	1,478 (77.5%)	1,592 (83.5%)	1,713 (89.8%)
Non-matching Instances	429 (22.5%)	315 (16.5%)	194 (10.2%)
Accuracy	97.12%	98.41%	98.73%
Precision	83.41%	88.59%	92.62%
Recall	77.5%	83.48%	89.83%
ALL-Reduced			
Matching Instances	1,410 (50.9%)	1,623 (58.6%)	1,609 (58.1%)
Non-matching Instances	1,359 (49.1%)	1,146 (41.4%)	1,160 (41.9%)
Accuracy	80.53%	85.04%	84.23%
Precision	60.87%	63.44%	60.99%
Recall	50.92%	58.61%	58.11%

creation or update. On an NJ tree layout, instances positioned far from the core of the tree (external instances), that is, placed on more external leaves, are the individuals that better characterize the class they belong to. On the other hand, instances positioned closer to the core of the tree (internal instances) represent the ones whose features do not fit well in any class, or that fit in more than one class. This is given by the nature of the NJ algorithm.

Three training sets are used, the first composed of external instances, the second composed of internal instances, and the third composed of a combination of the two previous ones. In all cases, training and test sets are disjoint.

For ALL-Reduced, 45 training instances were used, and the remaining 2,769 instances used as a test set. For ETHZ-Reduced, 112 instances were used, and the remaining 1,907 instances used as a test set. Table 3 shows the results of each classification for these collections. The worst results were obtained using the external instances to compose the training set. These are likely to be close to the centroids of their group, and unable to represent boundary elements of the class, resulting in a restrictive classifier. Using internal instances, information about the boundaries of the groups is fed to the classifier, promoting inclusion of a larger variety of features within each target class.

4.3 Constructing and Updating an LWPR Model

This experiment evaluates the use of visualization for the construction and update of an LWPR model to improve the classification results. In LWPR, a regression model, created from a set of training instances divided into C classes, is used to classify the data. It is possible to update this model by adding new instance information to reflect changes in the classification scenario. We investigated two usually employed classification strategies in the underlying PLS model: *One Against All* and *MulticlassMatrix* [30]. Our experimental results have shown no significant difference between the two approaches, considering computational cost and precision of the generated models. However, the One-Against-All approach requires the creation and updating of one model for each class, thus being more costly and spending more time to execute. This approach was employed in all the experiments presented in this paper. The constructed model is employed to classify the remaining instances of the collection.

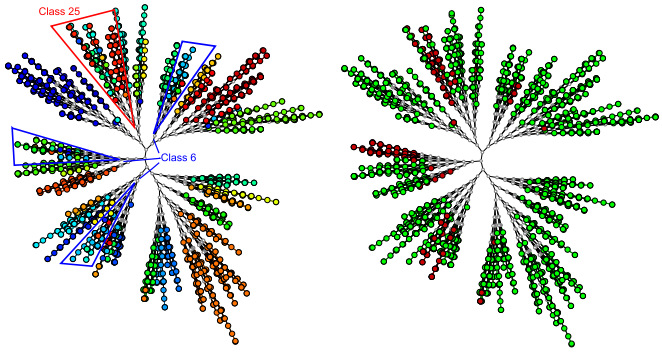
For ETHZ-Reduced, 84 training instances were used, three instances per class. The remaining 1,935 instances were used as a test set. For ALL-Reduced, 45 instances were used, five instances per class. The remaining 2,769 were used as a test set.

The numerical results of the classification for ETHZ-Reduced are presented in the second column of Table 4. A *ClassMatch* tool [29] was used to visually evaluate mismatch between ground truth and results, highlighting in red the misclassified instances. From this comparison, presented in Figs. 4a and 4b, as well as from the confusion matrix of this classification (not shown here due to its large size), one can notice that two classes concentrated the higher error rates, 6 (109 misclassified instances) and 25 (65 misclassified instances).

The layout provides several clues that may help to understand the structure of the collection, as well as classifier behavior. In Fig. 4a, it is possible to see that branches representing classes 6 and 25 are heterogeneous regarding class labels. As the layout is constructed based on similarity amongst instances, the branches are supposed to present a degree of homogeneity, specially in this experiment, where the dimensionality of the original collection was reduced focusing on class separability. If the branches are

TABLE 4
Results of ETHZ-Reduced and ALL-Reduced Classification Using the Initial and Updated LWPR Models

	Initial Model	Updated Model
ETHZ-Reduced		
Matching Instances	1,704 (88.1%)	1,808 (93.4%)
Non-matching Instances	231 (11.9%)	127 (6.6%)
Accuracy	98.47%	99.14%
Precision	89.05%	94.06%
Recall	88.06%	93.44%
ALL-Reduced		
Matching Instances	1,875 (67.7%)	1,991 (71.9%)
Non-matching Instances	894 (32.3%)	778 (28.1%)
Accuracy	86.61%	88.45%
Precision	71.98%	73.79%
Recall	67.71%	71.90%



(a) ETHZ-Reduced test tree. (b) *ClassMatch* - Initial model.

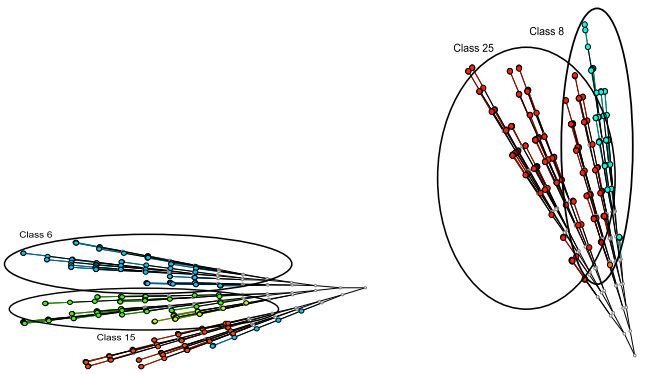
Fig. 4. Classified NJ tree of ETHZ-Reduced collection and correspondent *ClassMatch* tree, highlighting the heterogeneous branches corresponding the classification of instances originally from classes 6 and 25.

heterogeneous in terms of class labels, it may indicate that the classification is mixing instances in these branches.

Another clue presented by the tree of Fig. 4a is related to class 6, from which it is possible to see more than one branch. This may indicate that this class covers a wide range of features, being considerably heterogeneous, and that it could be divided into more homogeneous subclasses. For a class with such a complex structure, it is understandable that the correct classification of its instances is more difficult, and this can be confirmed by the numerical results.

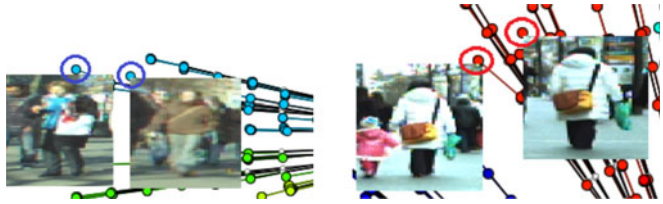
The analysis of the branches in the neighborhood of those representing classes 6 and 25 can help understand the classifier behavior. From the confusion matrix one can see that most of the instances from class 25 were classified as class 8, and most of the instances from class 6 were classified as class 15. The layout, in turn, shows that the branches corresponding to these classes are neighbors in the tree, as shown in Fig. 5, and in case of classes 25 and 8 (Fig. 5b), comprise one single branch, indicating that the instances of these classes are very similar.

The layout can also support users to create strategies to update the classification model. One possibility, using the *ClassMatch* tree, is to verify the branches with the highest misclassification rates, and search for features in the instances located in these branches that best describes the classes they belong. Possibly, the classifier is deficient in recognizing these features. It is important to notice that, in the absence of a ground truth, the user decides which points



(a) Classes 6 and 15. (b) Classes 25 and 8.

Fig. 5. Neighborhood of the branches corresponding classes 6 and 25, showing the similarity between instances of these classes.



(a) Class 6. (b) Class 25.

Fig. 6. Representative instances selected from classes 6 and 25.

are misclassified, and this misclassification is announced by points in close branches classified differently, or by heterogeneous branches, as stated before. Users can then select instances that present these features and use them to update the classifier. This strategy was employed in this experiment. In the tree, 23 misclassified instances, eight from class 25 and 15 from class 6, were selected after analysis of the *ClassMatch* tree branches. Fig. 6 presents some examples of this selection, showing the content of representative images from class 6 (Fig. 6a) and class 25 (Fig. 6b). These images were used to update the LWPR model, with the aim of reinforce the classifier knowledge about these classes. The third column of Table 4 shows the results of applying the updated model to a second classification.

The mismatches of the classification can be seen in Fig. 7, in red. Comparing this result with the *ClassMatch* tree presented in Fig. 4b, one can see an improvement for ETHZ-Reduced collection, also observed for ALL-Reduced collection.

To verify the role of the visualization in selecting instances for LWPR model update, we performed another experiment in which we reproduced the same steps performed above, but updating the training set with instances selected randomly. We employed 10 random sets for each data set, and the average classification obtained for ETHZ-Reduced was 1,802 (93.1 percent) matching instances, 132 (6.8 percent) non-matching instances, and accuracy, precision and recall rates of 99.06, 94.22 and 93.17 percent, respectively. For ALL-Reduced, the average classification result was 1,812 (65,4 percent) matching instances, 956 (34,5 percent) non-matching instances, and accuracy, precision and recall rates of 84.9, 74.19 and 65.45 percent, respectively.

For both collections, the improvement is better for the selection guided by the visualization. For ETHZ-Reduced, the improvement is not significant, which is expected, since this collection presents easier separability than the text data set. Since there are more selections likely to represent well the classes, improvement of the model is higher. On the other hand, ALL-Reduced presents worse class separability

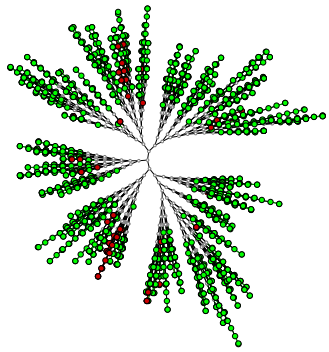


Fig. 7. Classification results using the updated LWPR model.

TABLE 5
Computational Time (in Seconds) to Create, Update and Apply the LWPR Models

Data set	ALL-Reduced	ETHZ-Reduced
Model Creation	0.391	1.804
Model Update	0.178	0.522
Model Application	1.210	1.819

and the criteria used for instance selection may considerably influence the model learning for each class. In this case, the layout played a crucial role to produce a satisfactory selection. Considering a real situation, in which there is no ground truth for a collection, the user is naturally the best resource to find a proper set of samples for model update, and the similarity based visual layout is a potentially valuable tool to perform the task.

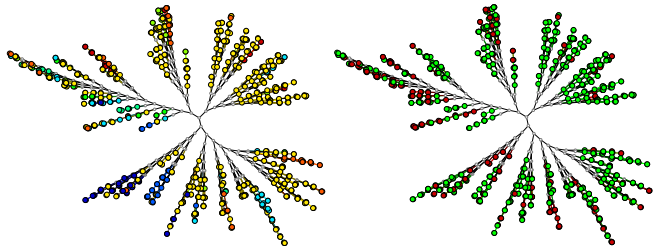
Table 5 shows the computational time required to construct, update and apply the LWPR models. These results show that VCM can provide an interactive and online adjustment procedure that allows for the user to adapt models to new scenarios in evolving collections. In all experiments, the time spent in the application of the initial model and all subsequent updated models did not presented significant difference. The time spent in each subsequent model update was also nearly constant, which shows that users do not experience significant performance impact as they iteratively update the model.

4.4 Iterative Classification

This experiment aims at verifying the convergence of a classification procedure using the application of a sequence of LWPR model updates. Initially, a LWPR model is built from a training set and used to classify a collection. Based on the result of this classification, the model is updated by the user and employed to classify a second collection. The model is updated again by using the results of the second classification, and another classification is performed on a third collection. This experiment was performed on ALL-Reduced collection and described as follows.

Iteration 1. From ALL-Reduced, three disjoints sets were built, an initial training set with 45 instances, and two other subsets, named *ALL-Reduced01* and *ALL-Reduced02*, with 926 and 922 instances, respectively.

The training set was employed to create an LWPR model that, in turn, was used to classify ALL-Reduced01 set. The numerical results of this classification is shown in the second column of Table 6 and the corresponding NJ tree and *Class Matching* tree presented in Fig. 8. Classes 2, 4, 8 and 0 presented high misclassification rates: 67.6, 56.3, 50.77 and 47.25 percent, respectively.



(a) NJ Tree. (b) *ClassMatch* Tree.

Fig. 8. NJ Tree of ALL-Reduced01 collection and corresponding classification result using the LWPR model created in iteration 1.

Iteration 2. Eight instances from classes 2, 4, 8 and 0 (2 from each classes) were selected to update the LWPR model, which in turn was used to classify ALL-Reduced02 set. The numerical results of this new classification are shown in Table 6, fifth column, and the corresponding NJ tree and *Class Matching* tree are presented in Fig. 9.

Table 6 shows the results of the first and second iterations of LWPR model applied to both ALL-Reduced01 and ALL-Reduced02 collections. It can be seen that updating the model improves classifications in both data sets.

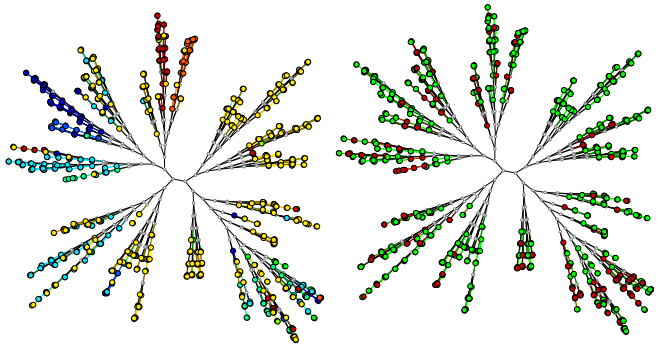
Iteration 3. Six misclassified instances from classes 2 and 3, that presented high misclassification rates, were selected from NJ tree of ALL-Reduced02 set and used to update LWPR model. This updated model was then used to classify a subset of the ALL-Reduced collection, with 2,769 instances, that contains the instances used in the model updates, but does not contain any instance of the initial training set. The results of the classification procedure using the three versions of the LWPR model are shown in Table 7. They show that the guided update provided by the tree layout supports robust convergence of the classifier.

4.5 Collection Evolution-New Classes

This experiment verifies how the visualization layout can assist the LWPR model update process when there is a change or evolution in the data set, and new classes appear. First, an LWPR model is created through a training set built from instances belonging to a classes and used to classify another collection containing instances of b classes, $a < b$. The performance of the classifier for instances belonging to known classes is examined, in search for possible unknown classes. Then, the model is updated using instances belonging to previously unknown classes and a new classification is performed. Two model update

TABLE 6
Classification Result Comparison Using LWPR Models Created in the Iterative Model Update Process on ALL-Reduced01 and ALL-Reduced02 Collection

	ALL-Reduced01		ALL-Reduced02	
Iteration	1	2	1	2
Matching Instances	632 (68.3%)	661 (71.4%)	613 (66.5%)	655 (71.0%)
Non-matching Instances	294 (31.7%)	265 (28.6%)	309 (33.5%)	267 (29.0%)
Accuracy	86.81%	88.40%	86.27%	88.22%
Precision	73.07%	73.99%	70.26%	73.10%
Recall	68.25%	71.38%	66.49%	71.04%



(a) NJ Tree. (b) *ClassMatch* Tree.

Fig. 9. NJ Tree of ALL-Reduced02 collection and corresponding classification result using the LWPR model created in iteration 2.

approaches were examined: the first uses only instances from unknown classes and the second uses a combination of instances belonging to known and unknown classes.

A subset of ETHZ-Reduced collection was used, composed of 717 instances organized into 10 classes, called *ETHZ-Reduced717*. From this subset, 100 instances from classes 4, 10, 16, 23, 25 and 26 were used to build an LWPR model. Classes 5, 7, 8 and 13 were not considered. Table 8 shows how the model classified instances from the six known classes, whereas Table 9 shows which classes the instances of unknown classes were inserted into.

Fig. 10 shows the ground truth of the collection (10a), as well as the *ClassMatch* tree from the classification procedure (10b). As expected, four branches were totally misclassified, representing the unknown classes.

Fig. 11 shows that instances from unknown class 8 are positioned at a branch closer to another branch that contains only instances from known class 25, possibly explaining why these instances were labeled to that class.

This example shows the layout capability to provide clues that potential new classes may be appearing in the collection, with instances represented by patterns that are unknown by the model. Here, two distinct branches present the same class label, indicating that one of them may be an unknown class, whose instances the model considered as a known class. Using multidimensional projections, the presence of partially or totally disconnected groups of instances, or even distant groups with the same class labels, may also indicate the appearing of new classes. These layout trends do not always represent new classes appearing in the collection, but they are clues that may indicate that a further analysis should be performed by the user.

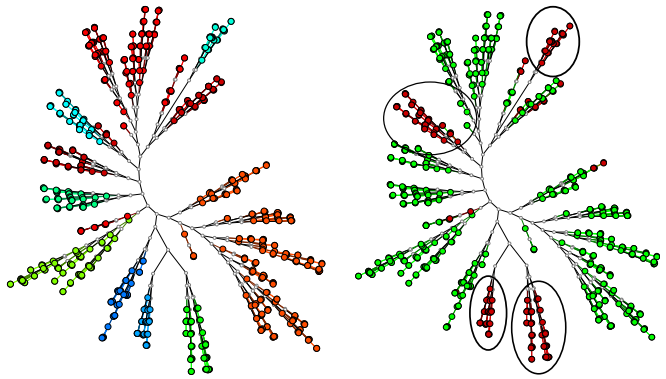
From this result, the LWPR model was updated through two strategies using only instances from unknown classes and using a combination of instances belonging to known and unknown classes. The numerical results of the classifications are shown in Table 10. The corresponding *Class Matching* trees are presented in Fig. 12. The results are better when the model is updated using instances that belong to known and unknown classes. When updated with only instances from the unknown classes, the model correctly classified all the instances from these classes, but as shown in the confusion matrix of Fig. 13, several instances from previously known classes, that were correctly classified before, were now misclassified.

TABLE 8
Performance of the Classifier for ETHZ-Reduced717 Collection on Instances of Known Classes

Class	Hit Rate
4	35/36 (97.2%)
10	47/47 (100.0%)
16	72/72 (100.0%)
23	207/211 (98.1%)
25	125/133 (93.9%)
26	57/73 (78.1%)

TABLE 9
Distribution of Instances from the Four ETHZ-Reduced717 Unknown Classes on the Six Known Classes by the LWPR Model

	4	10	16	23	25	26
5	15	3	0	8	3	0
7	1	11	10	0	7	13
8	0	0	0	0	27	0
13	2	21	1	5	9	9



(a) ETHZ-Reduced717 Ground truth. (b) Classification *ClassMatch* tree.

Fig. 10. Ground truth and *ClassMatch* tree comparison for ETHZ-Reduced717 using the model built from instances belonging to six classes.

TABLE 7
Classification Result Comparison Using Three Versions of the LWPR Model on ALL-Reduced Subset with 2,769 Instances

	ALL-Reduced Subset		
Iteration	1	2	3
Matching Instances	1,875 (67.7%)	1,946 (70.3%)	2,008 (72.5%)
Non-matching Instances	894 (32.3%)	823 (29.7%)	761 (27.5%)
Accuracy	86.61%	87.71%	88.24%
Precision	71.98%	72.84%	74.20%
Recall	67.71%	70.28%	72.52%

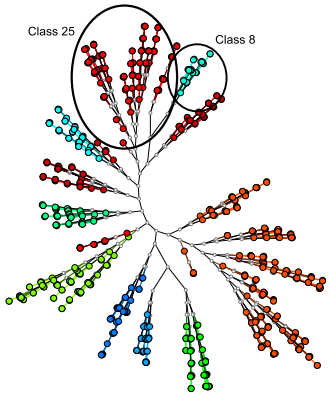
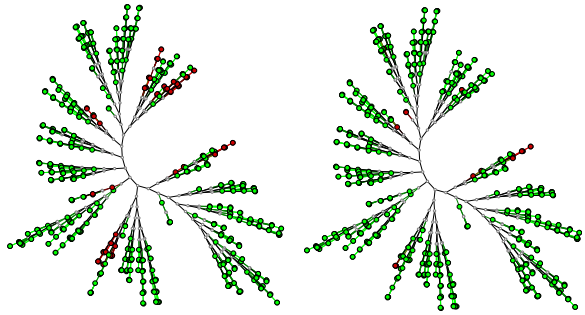


Fig. 11. NJ tree of ETHZ-Reduced717 highlighting the relationship between instances from classes 8 and 25.



(a) Update - 4 classes. (b) Update - 10 classes.

Fig. 12. ETHZ-Reduced717 *ClassMatch* trees using LWPR model updated with only unknown classes (12a) and with instances from all classes (12b).

4.6 Other Point Placements as Support to VCM

Naturally, similarity trees are not the only techniques that can support the visual classification process. Any 2D point placement of multidimensional data, such as multidimensional projections, could potentially be employed. This section illustrates and compares the use of a projection technique as an alternative to interact with data in the VCM.

An LSP projection and an NJ tree were constructed from the Free-Photo collection, and, from these layouts, sets of samples were chosen twice. First, a set of points is selected as input to a PLS supervised dimension reduction in order to improve feature space regarding class segregation. Then the resulting space it is mapped on screen, and again a set of points is chosen as samples to build the LWPR classification model. The model is used to classify the data set. Finally, new samples are chosen to update the classification model and the classification is performed again. All these steps are performed both with the projection and with the tree to verify their adequacy to sample set selection in the Visual Classification Procedure. This was a *one go procedure*, that is, there was no going back in the pipeline $selection \rightarrow reduction \rightarrow classification \rightarrow classification \quad improvement$. Also, a single improvement step was performed.

4.6.1 Characterization and Results of Classification Steps

Sample selection for supervised dimension reduction. The intention of this step was to select between 600 and 700 points that included samples of all nine classes. On the LSP layout, a set of 670 points was chosen (see Fig. 14a). Ten trials were made until all classes were sampled reasonably well. Then, in the tree layout, 610 points were chosen (see Fig. 14b). The difference in number is because the tree layout supported easy recognition of neighborhoods, so satisfaction with the set was achieved sooner. Five trials sufficed to reach the final sample set. Then, PLS dimension reduction (10 factors, target dimensionality 9) was employed on the original data set with each sample set separately. As a result, two reduced data sets of $3,462$ samples with nine attributes were built. We call the reduced space from the 670 points chosen from the projection 670_reduced, and the reduced space from the 610 points chosen from the NJ tree 610_reduced. Layouts of the reduced spaces can be seen in Fig. 16a and also in Figs. 16b and 16c. The employment of dimension reduction in fact improved the quality of the original space regarding discrimination of classes. This can be verified by the calculation of the silhouette coefficient, a measurement of cluster cohesion and separation [41] that varies between -1.0 and 1.0. Silhouettes of the original space, reduced spaces and their layouts can be verified by the curve in Fig. 15.

Sample selection for classification. After dimension reduction, the reduced space with better silhouette (610_reduced) was employed to carry on the rest of the work, that is, to classify the data set. The procedure was done by selecting samples and then submitting to LWPR for modeling and classification. Again, both layouts were employed to support the selection of samples for training the classifier. The target was again to choose between 600 and 700 points from the labeled data set. The interactions with the layouts were very similar to the ones done in the previous step, only now working on the reduced data set, that is, with layouts entailing better segregation (see Fig. 16). Therefore, in principle, location of well grouped points is easier. In total, 675 points from the projection layout and 598 points from the tree layout were chosen. A model for each training set was created (675-model and 598-model respectively). They were both used to classify the Free-Photo data set. The classification

TABLE 10
ETHZ-Reduced717 Classification Result Comparison Using LWPR Model Updated with Only Unknown Classes and with Instances from All Classes

	4 classes	10 classes
Matching Instances	640 (89.3%)	691 (96.4%)
Non-matching Instances	77 (10.7%)	26 (3.6%)
Accuracy	97.85%	99.00%
Precision	93.26%	97.25%
Recall	89.26%	96.37%

	4.0	5.0	7.0	8.0	10.0	13.0	16.0	23.0	25.0	26.0
4.0	16	20	0	0	0	0	0	0	0	0
5.0	0	29	0	0	0	0	0	0	0	0
7.0	0	0	42	0	0	0	0	0	0	0
8.0	0	0	0	27	0	0	0	0	0	0
10.0	0	0	0	0	47	0	0	0	0	0
13.0	0	0	0	0	0	47	0	0	0	0
16.0	0	0	0	0	0	0	72	0	0	0
23.0	0	8	0	3	0	3	0	197	0	0
25.0	0	0	0	13	0	0	2	1	117	0
26.0	0	6	0	1	0	13	2	0	5	46

Fig. 13. Confusion matrix associated to the ETHZ-Reduced717 classification using LWPR model updated with only unknown classes.

(a) Selection 1: 670 points selected.

(b) Selection 2: 610 points selected.

Fig. 14. Visual displays of the Free-Photo 3,462 points data set.

results were 73 percent of correct classification for the first and 75 percent for the latter.

Incremental classification. In this step, additional points were chosen from each layout using class match for projection and tree as guide. For the projection layout, 173 extra points were chosen and fed into the 675-model. From the NJ tree layout, 171 points were chosen and fed into the 598-model. The updated models were applied to the original data set and gave as result 76 and 78 percent correct classification, respectively.

4.6.2 Observations and Discussion of the Comparative Classification Experiment

From employing both projections and similarity trees as tools for the same classification process, a few observations can be drawn. The first one relates to the comfort finding appropriate samples. Locating cohesive groups from which

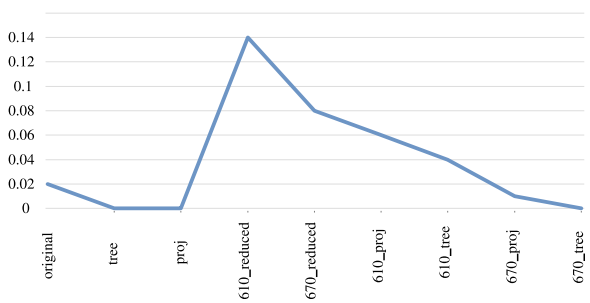
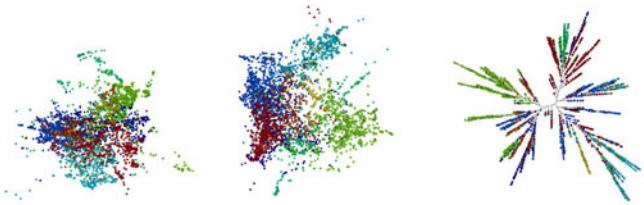


Fig. 15. Silhouette for original and reduced spaces, and layouts.



(a) 670_reduced. (b) 610_reduced. (c) 610_reduced.

Fig. 16. Layouts of the reduced data set by LSP and NJ tree.

to choose appropriate samples is at the core of successful user controlled sampling. Appropriate sampling would include points within those cohesive groups and at the boundary between groups. While groups of samples can be examined in either type of layout, via various tools for instance observation, the trees tend to place well adjusted individuals at the ends of branches, while, from the top of branches, one can select boundary individuals. Making these individuals easy to locate is one of the benefits of the NJ tree layout. Projections, on the other hand, have the problem of scale during interaction, which requires zooming in and out if one wants to understand whether there is separation of groups seemingly mixed, even when class discrimination is improved by dimension reduction. This is an issue for all projections, not only for LSP. This, however, does not prevent finding good samples, as can be seen from the similar final results.

While the quality of final results are similar when using either plot, the productivity and the quality of sample set selection were more favorable to the tree. The time to select points on the projection was larger due to the difficulty in estimating the density when selecting from cohesive groups. Too many or too few points were sometimes selected in equal areas of the visual space. The improvement in quality of the reduced space was larger for the reduction from the tree samples. Additionally, while the final classification results using the tree were only marginally better, fewer samples were necessary to achieve those results. Quality of the samples can be confirmed by plotting the sample sets (see Fig. 17). It can be seen from the figure that class segregation was better for the sample set chosen from the tree. Silhouettes of the layouts confirm that visual impression (Fig. 15).

Another aspect that favors the tree as the appropriate tool in this process is its complete absence of parameterization.

(a) Silhouettes 0.14/0.06. (b) Silhouettes 0.21/0.18.

Fig. 17. Projections of samples selected from each layout. (a) 675 points selected for classification from Projection View in 16b. (b) 598 points selected for classification from Tree View in 16c. Silhouettes are those of the feature space and of the layout.

The input for the tree is the similarity relationship or the feature space and the output is the tree. Projections, on the other hand, are usually subject to a series of parameters that tend to be very important in regards to quality of the final layout. In the case of LSP, its main parameter, the number of control points, was set do default (10 percent).

Projections are better than trees in other respects. If the shape of projected groups is meaningful for an application, then they should be used, since trees destroy any information that projections might keep on group shape.

5 Conclusions and Future Work

This work presented the *Visual Classification Methodology* for incremental classification tasks. It yields visual support to classification of evolving data sets by allowing user's interference, via similarity based visualizations, during supervised classification in an integrated form, promoting user's control over model building, application, evaluation and evolution. Extensive testing demonstrated that the association between users and automatic classification procedures through visualization techniques have great potential to support convergence to efficient classifiers, as well as for supporting adaptation of a classification scenario as the data set evolves.

Bidimensional point-based visualizations provide the support to most classification tasks including sample selection, manual labeling, model building, result verification and model updating. The same set of tools supports an optional pre-processing step involving dimension reduction. Additional attribute mappings were developed to support detailed analysis of the results and for feedback into the classification processes.

A system to instantiate the methodology is provided including various classification methods, such as PLS, SVM and LWPR. Other classification techniques and most point placement approaches can adapt to the proposed approach. The work also offers evidence of forms in which similarity trees, such as NJ trees, show the structure of groups in a way that allows interacting accurately along the classification steps. Multidimensional projections are also found useful for the same purposes.

The process and the underlying techniques, as well as the system developed with all the features of the methodology, are incremental and accommodate evolution of the models and of the data sets over time. The system is made available for general use.

As a future step, it would be interesting to further explore the characteristics of point based as well as other layouts that reflect the particularities of the collection and the classifier behavior, identifying how they can help users to interact faster with the classification. A comparison of these layouts against others created by different visualization approaches, with respect to their capabilities on composing strategies to update classification models, would also be an interesting research direction.

A limitation of the approach is the difficulties of interaction with very large data sets for both projections and trees, due to clutter. We are currently working on a version of the tree that is multi-scale and coupled with an improved way to use visual space (see [40]).

Acknowledgments

The authors are grateful to FAPESP, FAPEMIG, and CNPq (all Brazilian research support agencies).

References

[1] M. Agarwal, M. Goyal, and M. C. Deo, "Locally weighted projection regression for predicting hydraulic parameters," *Civil Eng. Environ. Syst.*, vol. 27, no. 1, pp. 71–80, 2010.

[2] C. Bachmaier, U. Brandes, and B. Schlieper, "Drawing phylogenetic trees," in *Proc. 16th Int. Conf. Algorithms Comput.*, 2005, vol. 3827, pp. 1110–1121.

[3] J. Choo, C. Lee, C. K. Reddy, and H. Park, "Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 1992–2001, Dec. 2013.

[4] J. Choo, H. Lee, J. Kihm, and H. Park, "iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction," in *Proc. IEEE Symp. Vis. Analy. Sci. Technol.*, 2010, pp. 27–34.

[5] G. Ciocca, C. Cusano, and R. Schettini, "Semantic classification, low level features and relevance feedback for content-based image retrieval," *Proc. IS&T/SPIE Electron. Imaging*, vol. 7255, p. 72550D, 2009.

[6] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*, 2nd ed. London,U.K.: Chapman & Hall, 2000.

[7] A. M. Cuadros, F. V. Paulovich, R. Minghim, and G. P. Telles, "Point placement by phylogenetic trees and its application for visual analysis of document collections," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, Sacramento, CA, USA, 2007, pp. 99–106.

[8] T.-N. Do, "Towards simple, easy to understand, an interactive decision tree algorithm," College Inf. Technol., Can tho Univ., Can Tho, Vietnam, Tech. Rep. 06-01, 2007.

[9] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Maui, HI, USA, 2001, vol. 1, pp. 298–303.

[10] P. A. Eades, "A heuristic for graph drawing," *Congressus Numerantium*, vol. 42, pp. 149–160, 1984.

[11] A. Ess, B. Leibe, K. Schindler, and L. van Gool, "A mobile vision system for robust multi-person tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, 2008, pp. 1–8.

[12] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Comput. Vis. Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[13] J. Florez, D. Bellot, and G. Morel, "LWPR-model based predictive force control for serial comanipulation in beating heart surgery," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Budapest, Hungary, 2011, pp. 320–326.

[14] G. M. Foody and A. Mathur, "The use of small training sets containing mixed pixels for accurate hard image classification: Training on mixed spectral responses for classification by a SVM," *Remote Sens. Environ.*, vol. 103, no. 2, pp. 179–189, 2006.

[15] F. Heimerl, S. Koch, H. Bosch, and T. Ertl, "Visual classifier training for text document retrieval," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2839–2848, Dec. 2012.

[16] B. Hoferlin, R. Netzel, M. Hoferlin, D. Weiskopf, and G. Heidemann, "Inter-active learning of ad-hoc classifiers for video visual analytics," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol.*, 2012, pp. 23–32.

[17] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Semi-supervised SVM batch mode active learning for image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–7.

[18] M. S. Hossain, P. K. R. Ojili, C. Grimm, R. Muller, L. T. Watson, and N. Ramakrishnan, "Scatter/gather clustering: Flexibly incorporating user feedback to steer clustering results," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2829–2838, Dec. 2012.

[19] I. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY, USA: Springer, 2002.

[20] A. J. Joshi, F. Porikli, and N. P. Papanikolopoulos, "Scalable active learning for multiclass image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2259–2273, Nov. 2012.

[21] D. A. Keim, C. Panse, and M. Sips, *Information Visualization: Scope, Techniques and Opportunities for Geovisualization*. New York, NY, USA: Elsevier, 2005.

[22] S. Klanke, S. Vijayakumar, and S. Schaal, "A library for locally weighted projection regression," *J. Mach. Learn. Res.*, vol. 9, pp. 623–626, 2008.
[23] P. A. Legg, D. H. 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 Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2109–2118, Dec. 2013.
[24] X. Li, R. Guo, and J. Cheng, "Incorporating incremental and active learning for scene classification," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, 2012, vol. 1, pp. 256–261.
[25] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognit.*, vol. 40, no. 1, pp. 262–282, 2007.
[26] M. Migut and M. Worring, "Visual exploration of classification models for risk assessment," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, 2010, pp. 11–18.
[27] A. Muthukumaravel, S. Purushothaman, and A. Jothi, "Implementation of locally weighted projection regression network for concurrency control in computer aided design," *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, pp. 46–50, 2011.
[28] G. Nguyen and M. Worring, "Interactive access to large image collections using similarity-based visualization," *J. Vis. Lang. Comput.*, vol. 19, no. 2, pp. 203–224, 2008.
[29] J. G. Paiva, L. Florian, H. Pedrini, G. Telles, and R. Minghim, "Improved similarity trees and their application to visual data classification," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 11, pp. 2459–2468, Dec. 2011.
[30] J. G. S. Paiva, W. R. Schwartz, H. Pedrini, and R. Minghim, "Semi-supervised dimensionality reduction based on partial least squares for visual analysis of high dimensional data," *Comput. Graph. Forum*, vol. 31, pp. 1345–1354, 2012.
[31] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz, "Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 3, pp. 564–575, May/Jun. 2008.
[32] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.
[33] R. O. Stehling and M. A. Nascimento, and A. X. Falcão, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *Proc. 11th Int. Conf. Inf. Knowl. Manage.*, 2002, pp. 102–109.
[34] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood, "Does organisation by similarity assist image browsing?" in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2001, pp. 190–197.
[35] S. Rüger, "Putting the user in the loop: Visual resource discovery," in *Proc. 3rd Int. Conf. Adaptive Multimedia Retrieval: User, Context, Feedback*, 2006, vol. 3877, pp. 1–18.
[36] W. R. Schwartz and L. S. Davis, "Learning discriminative appearance-based models using partial least squares," in *Proc. Brazilian Symp. Comput. Graph. Image Process.*, Rio de Janeiro, Brazil, 2009, pp. 322–329.
[37] W. R. Schwartz, H. Guo, J. Choi, and L. S. Davis, "Face identification using large feature sets," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2245–2255, Apr. 2012.
[38] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin, Madison, WI, USA, Tech. Rep. 1648, 2010.
[39] C. Shi, C. Xu, and X. Yang, "Study of TFIDF algorithm," *J. Comput. Appl.*, vol. 29, pp. 167–170, 2009.
[40] L. Tan, Y. Song, S. Liu, and L. Xie, "Imagehive: Interactive content-aware image summarization," *IEEE Comput. Graph. Appl.*, vol. 32, no. 1, pp. 46–55, Jan./Feb. 2012.
[41] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Boston, MA, USA: Addison-Wesley, 2005.
[42] E. Tejada, R. Minghim, and L. G. Nonato, "On improved projection techniques to support visual exploration of multidimensional data sets," *Inf. Vis.*, vol. 2, no. 4, pp. 218–231, 2003.
[43] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proc. 9th ACM Int. Conf. Multimedia*, 2001, pp. 107–118.
[44] Y. Tong, B. Safadi, and G. Quénot, "Incremental multi-classifier learning algorithm on Grid'5000 for large scale image annotation," in *Proc. Int. Workshop Very-Large-Scale Multimedia Corpus, Mining Retrieval*, 2010, pp. 1–6.
[45] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari, "A survey of active learning algorithms for supervised remote sensing image classification," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 606–617, Jun. 2011.
[46] S. van den Elzen and J. J. van Wijk, "BaobabView: Interactive construction and analysis of decision trees," in *Proc. IEEE Conf. Vis. Analy. Sci. Technol.*, 2011, pp. 151–160.
[47] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[48] S. Vijayakumar, A. D'Souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural Comput.*, vol. 17, pp. 2602–2634, 2005.
[49] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: An $O(n)$ algorithm for incremental real time learning in high dimensional space," in *Proc. Int. Conf. Mach. Learn.*, Stanford, CA, USA, 2000, pp. 1079–1086.
[50] M. Wang and X.-S. Hua, "Active learning in multimedia annotation and retrieval: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 2, p. 10, 2011.
[51] H. Wold, "Partial least squares," in *Encyclopedia of Statistical Sciences*, vol. 6. New York, NY, USA: Wiley, 1985, pp. 581–591.
[52] M. Worring, "Easy categorization of large image collections by automatic analysis and information visualization," in *Proc. Int. UDC Seminar Classification Vis.: Interfaces Knowl.*, 2013, p. 8.
[53] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia Syst.*, vol. 8, no. 6, pp. 536–544, 2003.

Jose Gustavo S. Paiva received the PhD degree from the University of Sao Paulo, Sao Carlos, Brazil. He is currently an assistant professor at the Federal University of Uberlandia, Uberlandia/MG, Brazil. His research interests include information visualization applied to image and text collections, and visual data classification, specifically the analysis of how the produced layouts may improve perception and comprehension of the classification process by users.

William Robson Schwartz received the BSc and MSc degrees in computer science from the Federal University of Parana, Curitiba, Brazil. He received the PhD degree in computer science from the University of Maryland, College Park. He is currently a professor in the Computer Science Department at Universidade Federal de Minas Gerais, Brazil. His research interests include computer vision, pattern recognition and, image processing.

Helio Pedrini received the BSc degree in computer science, and the MSc degree in electrical engineering, both from the University of Campinas, Brazil. He received the PhD degree in electrical and computer engineering from Rensselaer Polytechnic Institute, Troy, NY. He is currently a professor at the Institute of Computing at the University of Campinas, Brazil. His research interests include image processing, computer vision, pattern recognition, and computer graphics.

Rosane Minghim received the MSc degree in electrical engineering from the University of Campinas, Brazil, and the PhD degree in computer studies from the University of East Anglia, United Kingdom. She is an associate professor at the University of São Paulo, São Carlos, Brazil. She is interested in all aspects of visualization, information visualization, and visual analytics.