Semi-Supervised Learning with Interactive Label Propagation guided by Feature Space Projections

Bárbara C. Benato
Institute of Computing
University of Campinas
Campinas, Brazil
Email: barbarabenato@gmail.com

Alexandru C. Telea
Department of Mathematics and Computing Science
University of Groningen
Groningen, the Netherlands
Email: a.c.telea@rug.nl

Alexandre X. Falcão
Institute of Computing
University of Campinas
Campinas, Brazil
Email: afalcao@ic.unicamp.br

***Abstract*—While the number of unsupervised samples for data annotation is usually high, the absence of large supervised training sets for effective feature learning and design of high-quality classifiers is a known problem whenever specialists are required for data supervision. By exploring the feature space of supervised and unsupervised samples, semi-supervised learning approaches can usually improve the classification system. However, these approaches do not usually exploit the pattern-finding power of the user's visual system during machine learning. In this paper, we incorporate the user in the semi-supervised learning process by letting the feature space projection of unsupervised and supervised samples guide the label propagation actions of the user to the unsupervised samples. We show that this procedure can significantly reduce user effort while improving the quality of the classifier on unseen test sets. Due to the limited number of supervised samples, we also propose the use of auto-encoder neural networks for feature learning. For validation, we compare the classifiers that result from the proposed approach with the ones trained from the supervised samples only and semi-supervised trained using automatic label propagation.**

I. Introduction

Discriminative deep neural networks have been successful in image classification at the cost of processing many training examples per class. Typical applications in the science areas, such as medicine and biology, do not usually count on such large and pre-annotated datasets. Moreover, manual annotation of a large number of such training samples (e.g., images) by specialists from the application area is usually impractical due to high costs and their downtime.

Considering that the number of unsupervised (unlabeled) samples is usually much higher than the number of supervised (labeled) samples, semi-supervised learning methods have explored the data feature space to propagate labels to the unsupervised samples [1]–[7]. A classifier can then be trained from the resulting large set of labeled samples. Even the feature space can be redesigned by using that large labeled set to train discriminative deep neural networks. Additionally, there are methods that exploit generative deep neural networks for semi-supervised learning [8]–[14]. Such works, however, do not usually exploit the superior cognitive abilities of humans in recognizing patterns during the semi-supervised learning process. In contrast, crowd-sourcing tools rely on the knowledge of multiple users for manual annotation [15], [16] — an approach that could explore consensus and consistency analysis among users, if they acted during the machine learning process.

In order to obtain a large training set with accurately labeled samples, while keeping at minimum the user effort in data supervision, we propose a semi-supervised approach that exploits the superior pattern-finding power of the user's visual system to propagate labels to the unsupervised samples. The user is guided by two-dimensional t-SNE projections [17] of the feature space with the training samples, supervised and unsupervised ones. The use of such information visualization (infovis) technique to understand high-dimensional data and, more specifically, the results of machine learning techniques is not new [18]. Yet, the evidence that infovis and, more specifically, its interactive hypothesis-forming-and-validation sense making loop, known as Visual Analytics (VA), is effective to understand and improve machine learning [19]–[21], is very recent. These results, however, rely on a reasonable (in practice, large) number of supervised samples per category. The use of projections to improve clustering has also been proposed [22]. However, this approach only uses unsupervised samples and does not aim at data annotation.

In this work, we advocate a different, and to our knowledge novel, use of VA for supporting classification system engineering — the creation of large sets of labeled data that in turn support the construction of high-quality classifiers. Also, we explore the ability of Auto-encoder Neural Networks (AuNNs) to learn features from unsupervised samples and compare classifiers that result from the proposed method with others trained from the supervised samples only and semi-supervised trained using automatic label propagation. The results show that our method can usually obtain significant accuracy gains, better than the baselines, from little user effort. As main contribution, humans can recognize groups of samples with distinct shapes and limited class information in 2D better than machines can recognize them in nD

II. Proposed Pipeline

As stated earlier, the main question is how to support end users in creating large and labeled training sets with little effort such that this can provide high-quality classifiers? To answer this question, we split it into sub-questions that are addressed in turn. Figure 1 presents the workflows of the proposed visual

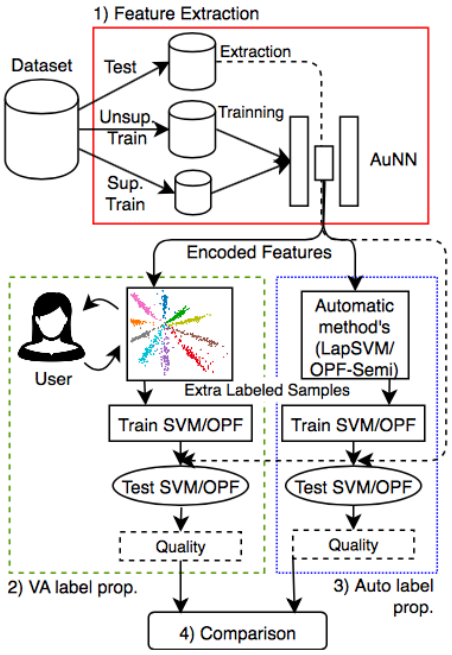

Fig. 1. The visual analytics workflow for interactive label propagation (dashed green box on the left) and the baseline workflow with automatic label propagation (solid blue box on the right). Both use the image features from an AuNN (the red box).

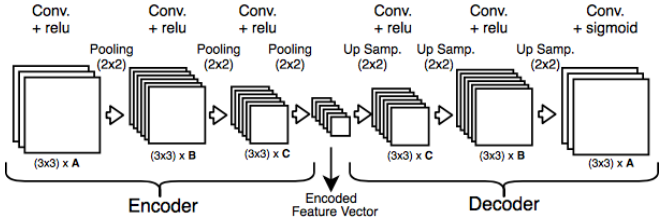

Fig. 2. AuNN architecture with convolutional layers used for image feature extraction. The convolutional layers have $A \geq B \geq C$ 3×3 dimensional filters as shown, whose values depend on the dataset. The convolutional layers are followed by ReLU activation and max-pooling in the encoder and, in the decoder, by ReLu activation and up-sampling, except the last layer that uses sigmoid activation.

analytics method based on interactive label propagation and of the baselines based on automatic label propagation. The individual steps are discussed in detail in the next sections.

1. Feature extraction: Given the limited number of supervised samples, this module relies on an Auto-encoder Neural Network (AuNN) [23] to learn the feature space from the training images, supervised and unsupervised ones. (Section III).

2. VA for interactive label propagation: The t-SNE algorithm [17] projects on 2D the created feature space with training samples to guide the interactive label propagation (ILP) (Section IV).

3. Automatic methods: The semi-supervised approaches, Laplacian Support Vector Machines (LapSVM) [1], [2] and Optimum-Path Forest (OPF-Semi) [5], are used to automatically propagate labels to the unsupervised samples (Section V).

4. Quality comparison: For both workflows, we compare the performance on unseen test sets of Support Vector Machine (SVM) [24] and Optimum-Path Forest (OPF) [25] classifiers trained with the large labeled set as well as with the supervised samples only (Section VI).

III. Feature Extraction

In order to keep at minimum the user effort, we assume the number of supervised samples is considerably less than the number of unsupervised samples in the training set. Thus, for a given dataset, we partition it into three subsets by using random and stratified sampling as follows: set S with 3% of supervised training samples, set U with 67% of unsupervised training samples (their labels are never used during training), and set T with 30% of unseen test samples. This procedure is also repeated three times for statistical analysis.

The training images in $S \cup U$ are submitted to the AuNN [23] for unsupervised feature learning. AuNNs consist of two modules: an *encoder* module that reduces the input image into a low-dimensional feature vector and a *decoder* module that reconstructs the input image from that feature vector [23], [26], [27]. The reconstruction errors are then minimized by adjusting the weights of the neural layers in the encoder and decoder during back-propagation. Different types of AuNNs could have been tried: fully-connected architectures with a single hidden layer [28]–[30], sparse networks with a regularization constraint added to the reconstruction error [31], and network architectures with a few hidden layers [26], [32]. We use an AuNN with convolutional layers due to their known success in shape and color feature extraction from images [23] (see Figure 2). This network is implemented in Python using Keras [33]. It contains three convolutional layers in the encoder module and three others in the decoder module, built using 3×3 dimensional filters. The convolutional layers are followed by *ReLu* activation and *max-pooling* in the encoder and by *ReLu* activation and 2×2 *up-sampling* in the decoder. The input images are normalized within $[0, 1]$ and, therefore, the last decoder layer uses a *Sigmoid* activation function, rather than *ReLu*, to produce a normalized reconstructed image.

This network architecture has been chosen experimentally on the training images by seeking a feature space after the encoder in which the sample projection on 2D by the t-SNE algorithm shows the structure of the data distributed into separated clusters. The assumption here is that those clusters are mostly populated by samples from a same class. Therefore, the projection of the supervised samples using a distinct color per class can guide the user to propagate the class labels to unsupervised samples.

IV. Interactive Label Propagation

The low-dimensional feature space created by applying the AuNN on training images can still present thousands of features. The use of non-linear projections on 2D seems

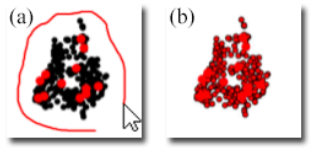

Fig. 3. The user (a) manually selects a sample group to (b) propagate labels to unsupervised samples.

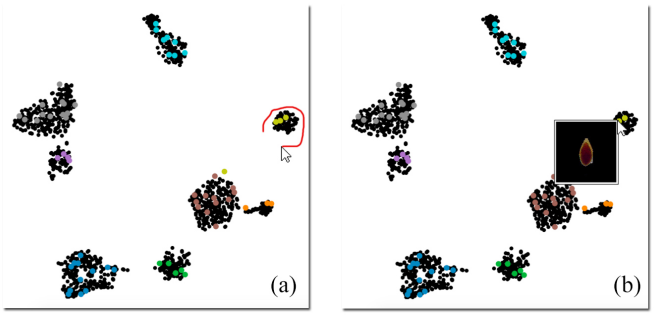

Fig. 4. (a) Projection of a training set with images of helminth eggs showing eight well-separated clusters. (b) The tooltips of the samples (small images) can be displayed to study the clusters.

to be the most suitable approach to understand such data distribution [32], [34]. Motivated by recent works [19]–[21] that indicate a strong relation between class separation in the high-dimensional feature space and the corresponding sample coordinates on 2D t-SNE projections [17], we adopted this method to guide the user's actions for label propagation. Moreover, it has been shown that the t-SNE algorithm can usually preserve the data distribution on the projection space better than other algorithms [35]. We have then used the t-SNE algorithm with its perplexity parameter fixed at 40 during 1,000 iterations.

For interactive label propagation (ILP), we developed a simple tool that starts by showing the t-SNE projection of $S \cup U$ with the class of the supervised samples in S color-coded by a categorical color map and the unsupervised samples in U shown in black. The visual identification of groups with a few samples from a same class and the remaining ones in back guides the user to propagate the label of that class to the unsupervised samples in that group. The extent of this label propagation is decided by manual free-sketching delineation using the mouse (see Figure 3). This label propagation is then limited to the most confident samples with respect to that class. It can be repeated multiple times and for any region of the projection. The tooltip (small image) is also displayed as the user moves the mouse over the corresponding 2D points (Figure 4). This cannot be considered label supervision, but it can quickly indicate whether the majority of the images in that group are from the same class and help the user to limit the extent of that label propagation.

V. Automatic Label Propagation

Semi-supervised learning techniques, such as LapSVM [1], [2] and Optimum-Path Forest [5], can explore the feature space with supervised and unsupervised samples to automatically assign labels to the unsupervised ones. SVM-based techniques usually rely on grid search with training images to determine its parameter values. However, due to the very limited number of supervised samples, we found better results by fixing the parameters of LapSVM as follows: kernel=*RBF*, kernel size=5, number of neighbors=6, $\gamma_A = 0.00001$, and $\gamma_I = 1.0$. The semi-supervised OPF (OPF-Semi) [5] does not have parameters. While LapSVM explores manifold regularization for label propagation, OPF-Semi interprets the training samples as nodes of a complete graph, computes a minimum-spanning tree in that graph, selects the supervised samples as seeds (prototypes), and then computes an optimum-path forest rooted at those prototypes, such that each unsupervised sample is assigned to the class of its most closely connected root in the feature space. The experiments then use these techniques to propagate labels from S to U.

VI. Quality Comparison

The success of the interactive and automatic label propagation processes can be measured by comparing the class of each sample in U and the propagated label. At the same time, it is desirable to train a classifier from the large labeled set $S \cup U$ such that its performance in assigning the correct label to the unseen test samples in T is high. Note, however, that errors in label propagation exist and so that training set $S \cup U$ may not be good enough to create an effective classifier. In order to evaluate the methods, we then use Support Vector Machine (SVM) [24] and Optimum-Path Forest (OPF) [25] as classifiers. We compare SVM trained on S – called "baseline" – with SVM trained on $S \cup U$, being U labeled by LapSVM and by ILP. We also compare OPF trained on S with OPF trained on $S \cup U$, being U labeled by OPF-Semi and by ILP.

When SVM is trained on $S \cup U$, its parameters are learned by grid search (3 splits with stratified random sampling) using 70% and 30% of the samples in $S \cup U$ for training and validation, respectively. OPF does not have parameters. Afterwards, the effectiveness of these classifiers was measured on T. As said, the whole process was repeated *three times*, as well as the label propagation, using random choices of S, U, and T, for statistical analysis, as discussed next. A single user with machine learning knowledge performed the ILP.

VII. Experiments and Results

We applied the pipeline described so far to several real-world datasets and classification problems. We next introduce the datasets and related classification tasks, discuss specific settings for our pipeline that depend on these datasets and tasks, present the effectiveness measures, and discuss the results.

A. Datasets

In order to validate the methodology, we first used the MNIST [36] public dataset. The MNIST dataset contains 28×28 dimensional images of handwritten digits from 0 to 9.

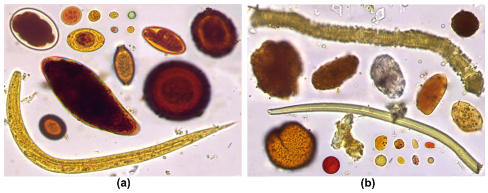

Fig. 5. Example of the parasites datasets: (a) parasites and its (b) impurities.

A random subset with $5,000$ images from the original training set with $60,000$ examples was chosen.

Next, we considered three medical image datasets. For these datasets, classical manual labeling (by actually viewing the images) would be both expensive and require a specialist user, so we argue that our VA label propagation is of added value in terms of cost saving. These datasets contain color images of human intestinal parasites already segmented from the background, centered, and resized to a 200×200 pixel resolution.

The datasets are separated in different stages: (i) *Helminth larvae*, (ii) *Helminth eggs* and (iii) *Protozoan cysts*. Besides these three classes, a fourth one exists: impurities. These are visually very similar to parasites (see example in Figure 5). Dataset (i) has $3,514$ images of two categories: helminth larvae and impurities. Dataset (ii) has $5,112$ images of nine categories: *H.nana*, *H.diminuta*, *Ancilostomideo*, *E.vermicularis*, *A.lumbricoides*, *T.trichiura*, *S.mansoni*, *Taenia*, and impurities. Dataset (iii) has $9,568$ images of seven categories: E.coli, *E.histolytica*, *E.nana*, *Giardia*, *I.butschlii*, B.hominis, and impurities. All three datasets are unbalanced. We also considered the datasets (ii) and (iii) without the impurity category for the purpose of exploring different levels of difficulty. Table I shows the number of images of each type in the considered datasets, considering also the split percentages for S, U and T defined in Section III.

B. Autoencoder Neural Network Set-up

For the MNIST dataset, the convolutional layers present 16, 8, 8, 8, 8 and 16 filters. For the Parasites datasets, we use 32, 16, 8, 8, 16 and 32 filters respectively. As a cost function to optimize, we considered both mean squared error and binary cross entropy, the latter giving better results with fewer training epochs, *i.e.* 50 for the easier datasets (MNIST and Helminth Eggs without impurity) and 100 for all other datasets, respectively. For MNIST, the feature vector has 128 dimensions. For the Parasites database, the feature vector has $5,000$ dimensions.

TABLE I
NUMBER OF SUPERVISED AND UNSUPERVISED TRAIN SAMPLES AND TEST SAMPLES.

Database		$\mid S \mid$	$\mid U \mid$	$\mid S \cup U \mid$	$\mid T \mid$
MNIST		175	3325	3500	1500
Larvae	with impurity	122	2337	2459	1055
Egg	w/o impurity	61	1176	1237	531
	with impurity	178	3400	3578	1534
Proto	w/o impurity	134	2562	2696	1156
	with impurity	334	6363	6697	2871

C. Effectiveness measures

The Cohen's kappa coefficient [37], [38] is more suitable to measure effectiveness in the case of unbalanced datasets, such as those used in this work. Thus, the performance of each method on the test sets is measured by the Cohen's kappa coefficient

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{1}$$

where p_o is the simple accuracy and

$$p_e = \frac{1}{N^2} \sum_k n_{k\alpha} n_{k\beta}, \tag{2}$$

where k is the number of categories, N is the number of samples, and $n_{k\alpha}$ and $n_{k\beta}$ are the predicted category k given by classifiers α and β, respectively. The κ coefficient is in a $[-1, 1]$ range, where $\kappa \leq 0$ means no agreement and $\kappa = 1$ means complete agreement between two classifiers α and β.

Apart from the above, we also compute the accuracy of the label propagation from S to U for the three label propagation methods tested (ILP, LapSVM, and OPF-Semi). We performed all computations (the entire workflow in Figure 1) three times to obtain more reliable statistics on our results.

We next present the results of the experiments, in increasing order of difficulty (MNIST dataset, Parasites dataset without impurities, and Parasites dataset with impurities).

D. MNIST dataset results

Figure 6 shows the t-SNE projection for the MNIST. As visible, smaller clusters have one predominant label color only, while larger ones contain more label values (colors). The user can reasonably easily 'split' these groups to propagate labels to the most confident samples.

Table II shows the classification results, the propagation accuracy and the number of labeled samples by each approach, averaged over three executions of the entire pipeline. The average Kappa for the baseline experiment considering the SVM and OPF classifiers shows a good degree of agreement. For

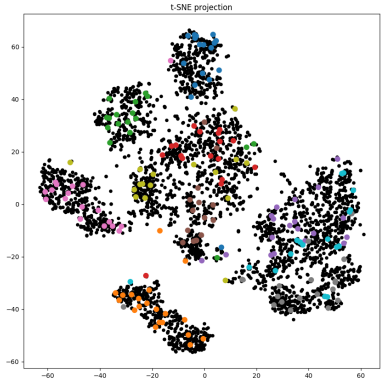

Fig. 6. $2D$ t-SNE projection of the MNIST dataset, with the label set $S \cup U$ color-coded by label values and the points in U drawn in black.

the SVM results, the label propagation done by ILP surpasses the baseline and LapSVM and OPF-Semi. In addition, note that even if the user propagates labels only to roughly 50% of the U, this performs better than when propagating samples automatically to *all* labels.

E. Parasites dataset results

Figure 7 shows the t-SNE projections for the Helminth eggs and Protozoan cysts datasets without impurities. The projection in Figure 7a allows to identify one consistent group per each category (label value) of the Helminth eggs. In Figure 7b, groups are much more mixed *vs* colors. This tells us that the Protozoan cysts dataset is more challenging than the Helminth eggs one, even when impurities are not considered.

Table III shows the classification results and label propagation accuracies for these two datasets. For the Helminth eggs, the average Kappa for SVM and OPF classifiers achieves results close to 1, *i.e.*, a high inter-rate of agreement. The ILP performs, again, better than the baseline for the SVM and OPF classifiers, with roughly the same number of annotated samples as the automatic propagation methods. Interestingly, LapSVM performs worse than the baseline for both classifiers. For the Protozoan cysts datasets, the ILP also gets better average Kappa than the baseline for the SVM and OPF classifiers. The LapSVM achieves a Kappa value less than 0.4 for both classifiers, which means fair agreement, while the other label propagation methods yield a Kappa of 0.7. This is in line with the perceived difficulty of this dataset reflected in Figure 7. Note also that the user propagates to only roughly 80% of the samples and obtains the best classification result.

Figure 8 shows the projections for the Helminth larvae, Helminth eggs, and Protozoan cysts datasets with impurities. For the Helminth larvae, we see that, although there are only two categories, a high mixture exists. Also, most of the impurities are located in the larger group (Figure 8b, light blue). Finally, the Protozoan cysts projection (Figure 8c) shows almost no separated groups, high label mixing, and an uniform spread of impurities (light blue). As such, we find this to be our most challenging (and actually also largest) dataset.

Table IV shows the classification results and propagation accuracies for the datasets in Figure 8. For the Helminth larvae, the Kappa coefficient obtained by the baseline is less than 0.55 for the SVM and OPF classifiers. The label propagation made by ILP yields a Kappa of 0.7 for both classifiers, thus surpassing the automatic label propagation methods. Even though these datasets are quite challenging, as discussed earlier and visible in Figure 8, the ILP propagated the labels with an accuracy of 98%. For the Helminth eggs dataset, the ILP gets the Kappa coefficient less than the baseline for 2% with the SVM classifier, but gets an increase of 3% with the OPF classifier. Again, LapSVM yields a Kappa value worse than the baseline for both classifiers. For the Protozoan cysts dataset, the ILP yields a Kappa value for the SVM and OPF classifiers which is less than the baseline by 4% and 0.2% respectively; however, it surpasses the automatic label propagation methods for both classifiers. We can see that this is the most difficult dataset reflected in all these results.

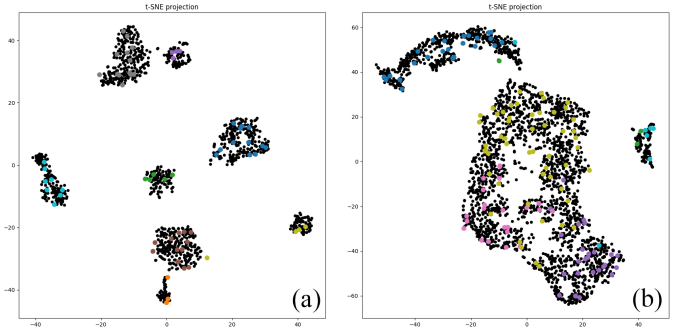

Fig. 7. t-SNE projection showing the $S \cup U$ sets of the (a) Helminth Eggs and (b) Protozoan cysts datasets without impurities.

VIII. Discussion

Let us revisit our proposed workflow and discuss its strong points and limitations.

A. User labeling added-value

The experiments realized in Section VII aim to compare the ILP with two automatic methods. For all considered datasets, in increasing level of difficulty, starting with MNIST and ending with the Parasites with impurities, we found that the user-based label propagation achieves in the end better classification results than the automatic label propagation methods considered, and also better than a classifier trained without any label propagation. As such, the added value of user-driven label propagation is justified.

B. Way of working

The proposed VA procedure is quite simple and does not require special training: The user sees a color-coded projection and is guided by the perceived shapes, clusters, and distances to decide where from, and how far, to propagate labels. Typically, one starts in the 'easy' areas showing compact groups containing only labels of the same class (color). One stops either when the allocated time (effort) for labeling has expired, or when the remaining samples are located in too complex (mixed) regions.

Understanding when to *stop* propagating is an interesting question: Propagating too little will yield limited added-value; propagating too far may create wrong labels, thus decrease the quality of the final classifier and also waste effort. To understand this process, we recorded the order in which the user labeled the Helminth larvae dataset, and plotted the increase or decrease of quality (measured by Kappa) and also the increase or decrease of propagation accuracy as a function of the number of added samples. Figure 9 shows these results.

Several insights appear from Figure 9. First, we see that the two considered classifiers (OPF and SVM) behave almost identically, so, the user should not be concerned by this choice during label propagation. Second, the increase of quality is not linear with the labeling effort. We see that a "saturation" effect

TABLE II
AVERAGE KAPPA AND ITS STANDARD DEVIATION FOR THE SVM AND OPF CLASSIFICATION RESULTS ON THE T SET OF MNIST DATASET. THE BEST RESULTS FOR EACH COLUMN OF QUALITY ARE IN BOLD.

Technique	$\mid S \mid$	Average $\mid U \mid$	Average Propagation Accuracy	Average $\mid S \cup U \mid$	Average Kappa (SVM)	Average Kappa (OPF)
baseline	175	-	-	175	0.813415 $\pm$ 0.001	0.709450 $\pm$ 0.021
LapSVM	175	3325	0.095639	3500	0.000000 $\pm$ 0.000	0.051110 $\pm$ 0.006
OPF-Semi	175	3325	0.763308	3500	-	0.721600 $\pm$ 0.043
ILP	175	1864	**0.974718**	2039	**0.844264 $\pm$ 0.027**	**0.776241 $\pm$ 0.036**

TABLE III
AVERAGE KAPPA AND ITS STANDARD DEVIATION FOR THE SVM AND OPF CLASSIFICATION RESULTS ON THE T SET OF HELMINTH EGGS AND PROTOZOAN CYSTS WITHOUT IMPURITIES.THE BEST RESULTS FOR EACH COLUMN OF QUALITY AND DATASET ARE IN BOLD.

Database	Technique	$\mid S \mid$	Average $\mid U \mid$	Average Propagation Accuracy	Average $\mid S \cup U \mid$	Average Kappa (SVM)	Average Kappa (OPF)
Helminth Eggs	baseline	61	-	-	61	0.961366 $\pm$ 0.023	0.941358 $\pm$ 0.026
	LapSVM	61	1176	0.886338	1236	0.873472 $\pm$ 0.035	0.877344 $\pm$ 0.037
	OPF-Semi	61	1176	0.947563	1236	-	0.939834 $\pm$ 0.051
	ILP	61	1171	**0.996014**	1232	**0.986624 $\pm$ 0.009**	**0.987364 $\pm$ 0.003**
Protozoan cysts	baseline	134	-	-	134	0.823106 $\pm$ 0.016	0.762682 $\pm$ 0.008
	LapSVM	134	2562	0.521598	2696	0.346761 $\pm$ 0.001	0.371770 $\pm$ 0.005
	OPF-Semi	134	2562	0.802238	2696	-	0.729438 $\pm$ 0.052
	ILP	134	1999	**0.947177**	2133	**0.851948 $\pm$ 0.006**	**0.841023 $\pm$ 0.002**

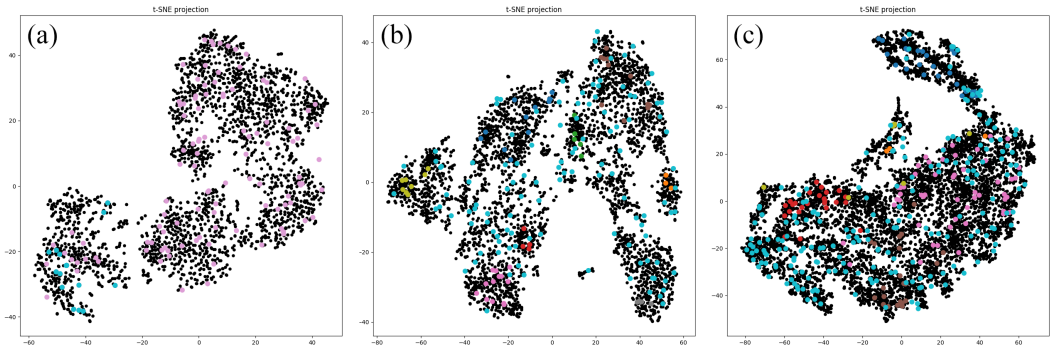

Fig. 8. t-SNE projection color-coded for the $S \cup U$ sets of the (a) Helminth larvae, (b) Helminth eggs and (c) Protozoan cysts datasets with impurities (the latter are colored light blue).

is reached at around 1900 added labeled samples. Adding more samples does not increase quality, and actually the propagation accuracy also drops slightly. This is because the used visual metaphor to propagate labels (projection) favors propagating from the "easy cases" (not mixed groups) first. When these cases are exhausted, only the complex (mixed and confusing) regions remain. From this point, further propagation is likely of limited or even negative value.

C. Effectiveness as a function of the data

It is also interesting to study how the classification quality and propagation accuracy correlate with the type of datasets being treated. For this, we plot the average Kappa and propagation accuracy per dataset, for our six studied datasets, for all the three considered propagation techniques (LapSVM, OPF-Semi, and ILP). Figure 10 shows these results for the OPF classifier, with the datasets sorted along the x axis on decreasing values of the quality of classifiers trained with user-propagated labels.

Several insights appear from this figure. First, we see that the ILP consistently beats the automatic propagation for all datasets. More interestingly, we see that this is more pronounced for the *difficult* datasets, *i.e.*, the ones for which the Kappa values are the smallest (shown to the right of the figure). Also, we see that the quality is correlated positively with the propagation accuracy – higher accuracies lead to a higher classification quality. Last but not least, the chart reflects an ordering of the datasets from easy to challenging which is in line with our own insights and what the projections shown in the earlier figures tell us. All in all, we see that manual propagation is of clear added value, *and* this value is relatively larger for complex datasets.

D. Scalability

In our experiments, we used datasets ranging from roughly $1,000$ to $7,000$ samples in the projection (set $S \cup U$), see Table I. Significantly larger datasets, *e.g.* having tens of thousands of samples, can pose problems as the projection

TABLE IV
AVERAGE KAPPA AND ITS STANDARD DEVIATION FOR THE SVM AND OPF CLASSIFICATION RESULTS ON THE T SET OF HELMINTH EGGS AND PROTOZOAN CYSTS WITH IMPURITIES. THE BEST RESULTS FOR EACH COLUMN OF QUALITY AND DATASET ARE IN BOLD.

Database	Technique	$\mid S \mid$	Average $\mid U \mid$	Average Propagation Accuracy	Average $\mid S \cup U \mid$	Average Kappa (SVM)	Average Kappa (OPF)
Helminth Larvae	baseline	122	-	-	122	0.375378 ± 0.333	0.531080 ± 0.035
	LapSVM	122	2337	0.882613	2459	0.121253 ± 0.086	0.173416 ± 0.088
	OPF-Semi	122	2337	0.920696	2459	-	0.600475 ± 0.071
	ILP	122	2080	**0.981273**	2202	**0.727843 ± 0.013**	**0.723049 ± 0.016**
Helminth Eggs	baseline	178	-	-	178	0.705972 ± 0.037	0.568304 ± 0.034
	LapSVM	178	3400	0.654118	3578	0.000000 ± 0.000	0.076043 ± 0.016
	OPF-Semi	178	3400	0.504510	3578	-	0.392956 ± 0.021
	ILP	178	1547	**0.914358**	1725	**0.683544 ± 0.033**	**0.593104 ± 0.034**
Protozoan cysts	baseline	334	-	-	334	**0.628584 ± 0.024**	**0.476051 ± 0.010**
	LapSVM	334	6363	0.622662	6697	0.232800 ± 0.104	0.202826 ± 0.030
	OPF-Semi	334	6363	0.468804	6697	-	0.339095 ± 0.033
	ILP	334	1787	**0.826867**	2121	0.589643 ± 0.036	0.472148 ± 0.008

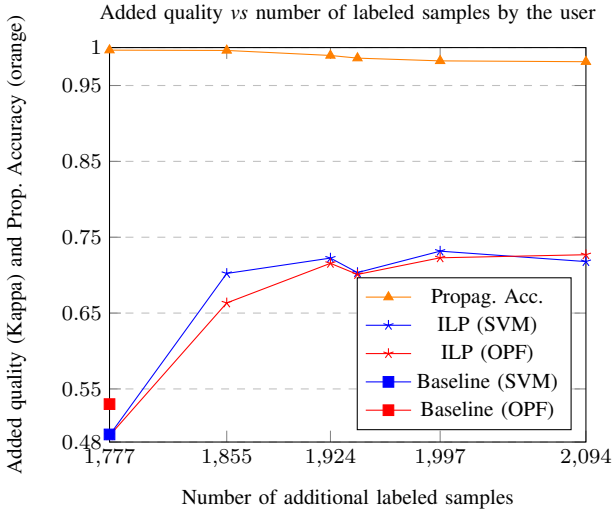

Fig. 9. Experiment presents the added quality by the number of additional labeled samples for the Helminth Larvae dataset during the ILP. Between the brackets at the legend, it is indicate the type of classifier (SVM or OPF) that was constructed given the propagated labels. The baseline quality is given by the squares. It is also shown the propagation accuracy.

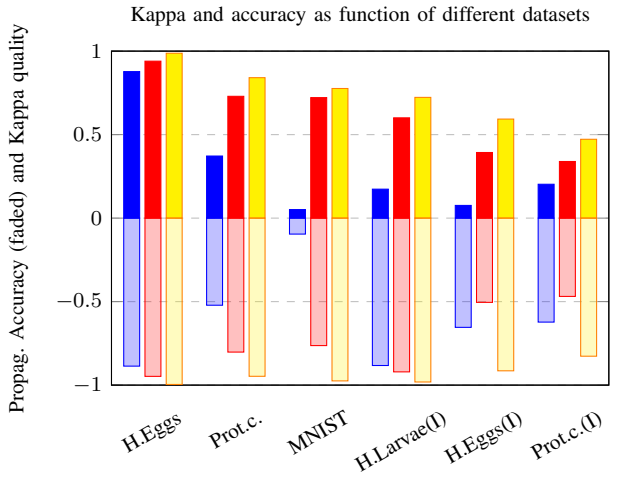

Fig. 10. Quality and propagation accuracy for six studied datasets *vs* three label propagation methods (LapSVM=blue, OPF-Semi=red, ILP=yellow). The top bars show quality (Kappa). The bottom bars show propagation accuracy.

will become cluttered, leading in turn to potential propagation errors.

E. Limitations

While demonstrably effective, our proposed VA approach has several limitations. As already noted, the projection metaphor becomes cluttered around $10,000$ samples, potentially leading to propagation errors. Also, we have shown that propagation becomes ineffective or even undesired after a certain threshold. Currently, this threshold is determined by the user based on the perceived difficulty of the visual patterns shown in the projection. Designing VA mechanisms to assist the user both in deciding the order in which to propagate and when to stop based on the data characteristics is a high-potential idea for future work. Separately, the number of displayed classes is limited inherently by categorical color coding to around 10. Overcoming this limit is an open problem in information visualization. However, solutions can be explored *e.g.* based on the fact that not all labels need to be shown at the same time to support label propagation.

IX. CONCLUSION

In this paper, we addressed the problem of sparsely-annotated datasets for classifier engineering by using a visual analytics based approach that leverages the pattern-finding power of the human eye to 'fill in' gaps in annotated visualizations. For this, we construct a projection of an image dataset by using autoencoder networks to extract features, and next project the feature space on a 2D scatterplot using t-SNE. Next, we let the user propagate labels (from a small set of existing ones) directly in the projection space. Finally, we use the augmented labeled set to train and test classifiers that use the extracted features.

While the actual VA technique being used to propagate labels is very simple, its end-to-end results are surprisingly good: Manual label propagation, even subject to errors done by the human user, achieves in the end consistent and better

classification performance than two modern automatic label propagation methods, for two different classifier techniques, over a collection of datasets and classification tasks ranging from simple to complex. This suggests that adding more support for the user during label propagation would only increase the quality of the obtained results even further.

We next plan to address precisely this last goal, by designing new visual mechanisms to inform and support the user during label propagation by using the nearest neighbors of a given point in nD space. Thus, we expect to maximize the quality of the obtained labels, minimize the propagation effort, and overall make the entire propagation process more transparent.

ACKNOWLEDGMENT

The authors are grateful to FAPESP grants #2014/12236-1, #2016/25776-0 and #2017/25327-3, and CNPq grants 302970/2014-2.

REFERENCES

[1] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: From transductive to semi-supervised learning," in *Proceedings of the 22Nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 824–831.

[2] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Dec. 2006.

[3] J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma, "Image annotation via graph learning," *Pattern Recognition*, vol. 42, no. 2, pp. 218 – 228, 2009, learning Semantics from Multimedia Content.

[4] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation," in *2009 IEEE 12th International Conference on Computer Vision*, Sept 2009, pp. 309–316.

[5] W. P. Amorim, A. X. Falcão, J. a. P. Papa, and M. H. Carvalho, "Improving semi-supervised learning through optimum connectivity," *Pattern Recogn.*, vol. 60, no. C, pp. 72–85, Dec. 2016.

[6] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, Jan 2008.

[7] J. Tang, R. Hong, S. Yan, T.-S. Chua, G.-J. Qi, and R. Jain, "Image annotation by knn-sparse graph-based label propagation over noisily tagged web images," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 2, pp. 14:1–14:15, Feb. 2011.

[8] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3581–3589.

[9] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *CoRR*, vol. abs/1511.06390, 2015. [Online]. Available: http://arxiv.org/abs/1511.06390

[10] M. A. Ranzato and M. Szummer, "Semi-supervised learning of compact document representations with deep networks," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 792–799.

[11] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 151–161.

[12] D.-H. Lee, "Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks," 07 2013.

[13] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 3546–3554.

[14] A. Gogna and A. Majumdar, "Semi supervised autoencoder," in *Neural Information Processing*, A. Hirose, S. Ozawa, K. Doya, K. Ikeda, M. Lee, and D. Liu, Eds. Cham: Springer International Publishing, 2016, pp. 82–89.

[15] L. von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '04. New York, NY, USA: ACM, 2004, pp. 319–326.

[16] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 157–173, May 2008.

[17] L. V. D. Maaten, "Accelerating t-SNE using tree-based algorithms," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[18] E. Packer, P. Bak, M. Nikkil, V. Polishchuk, and H. J. Ship, "Visual analytics for spatial clustering: Using a heuristic approach for guided exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2179–2188, Dec 2013.

[19] P. Rauber, A. Falcão, and A. Telea, "Projections as visual aids for classification system design," *Information Visualization*, 2017.

[20] P. Rauber, A. Falcão, and A. Telea, "Visualizing time-dependent data using dynamic t-sne," in *Proceedings of the Eurographics / IEEE VGTC Conference on Visualization: Short Papers*, ser. EuroVis '16, 2016, pp. 73–77.

[21] M. D. Zeiler and R. Fergus, *Visualizing and Understanding Convolutional Networks*. Cham: Springer International Publishing, 2014, pp. 818–833.

[22] D. Cohn, R. Caruana, and A. Mccallum, "Semi-supervised clustering with user feedback," Cornell University, Tech. Rep., 10 2003.

[23] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, *Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 52–59.

[24] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, July 1998.

[25] J. P. Papa, A. X. Falcão, V. H. C. de Albuquerque, and J. M. R. Tavares, "Efficient supervised optimum-path forest classification for large datasets," *Pattern Recognition*, vol. 45, no. 1, pp. 512 – 520, 2012.

[26] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes." *CoRR*, vol. abs/1312.6114, 2013.

[28] D. H. Ballard, "Modular learning in neural networks," in *Proceedings of the Sixth National Conference on Artificial Intelligence*, K. Forbus and H. Shrobe, Eds. San Francisco, CA: Morgan Kaufmann, 1987, pp. 279–284.

[29] Y. LeCun, "Modeles connexionnistes de l'apprentissage (connectionist learning models)," Ph.D. dissertation, Université P. et M. Curie (Paris 6), June 1987.

[30] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," Philips Research Laboratory, Brussels, Belgium, Manuscript M217, 1987.

[31] M. Ranzato, C. S. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model." in *NIPS*, B. Schölkopf, J. C. Platt, and T. Hofmann, Eds. MIT Press, 2006, pp. 1137–1144.

[32] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[33] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[34] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[35] "Dimensionality reduction: A comparative review."

[36] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[37] J. L. Fleiss and J. Cohen, "The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability," *Educational and Psychological Measurement*, vol. 33, no. 3, pp. 613–619, 1973.

[38] N. C. Smeeton, "Early history of the kappa statistic," *Biometrics*, vol. 41, pp. 795–795, 1985.