Reproducing Relative Attributes

Midterm Update

Joe Ellis

03/13/2012

Replication Goals:

1. Completely reproduce the proposed zero-shot learning framework from the paper titled "Relative Attributes", that appeared in the International Conference on Computer Vision (2011). Replication does not include the baselines that that author's used, because these are not the subject of the paper.
2. Recreate the graphs that can be seen below in Figure 1, and attempt to replicate the green line which is their proposed method.
3. Build a completely automated test bench so that my experiments can be run completely automatically with little to no knowledge of the work.

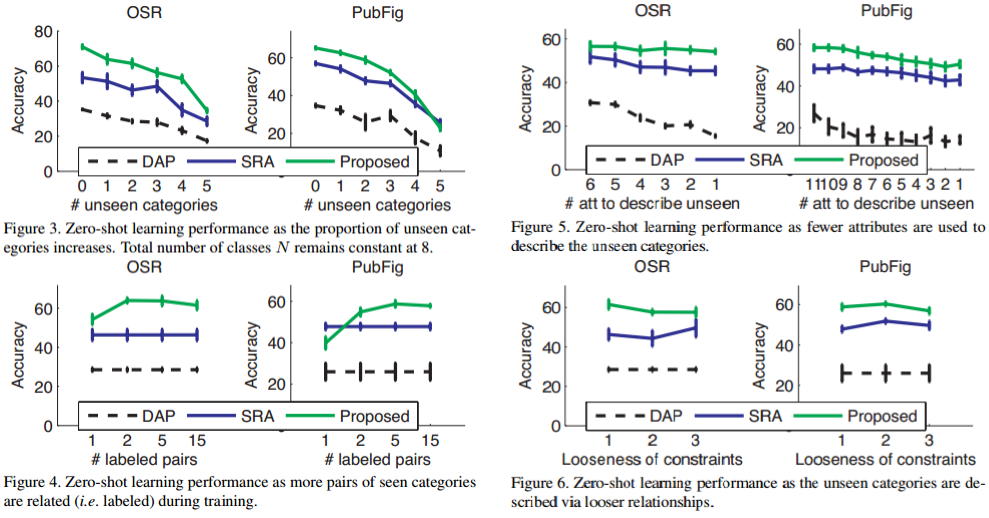

Figure 3. Zero-shot learning performance as the proportion of unseen categories increases. Total number of classes N remains constant at 8.

Figure 4. Zero-shot learning performance as more pairs of seen categories are related (*i.e.* labeled) during training.

Figure 5. Zero-shot learning performance as fewer attributes are used to describe the unseen categories.

Figure 6. Zero-shot learning performance as the unseen categories are described via looser relationships.

Figure 1. Results for Replication

Replication Results:

The current progress in replicating these results can be seen below in Figures 2-5. Figure 1 encompasses all the computational results from the figures in the paper, and each figure within Figure 1 is referenced below as Figure 1.[figure number]. The content of these figures will be expanded upon in the next section, which details the process that I have currently completed in attempting to replicate these results.

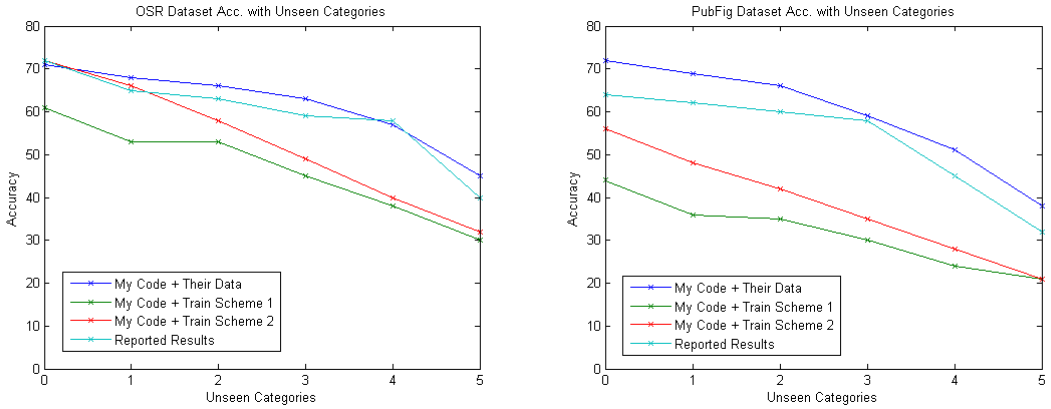

Figure 2. Replication results of Figure 1.3

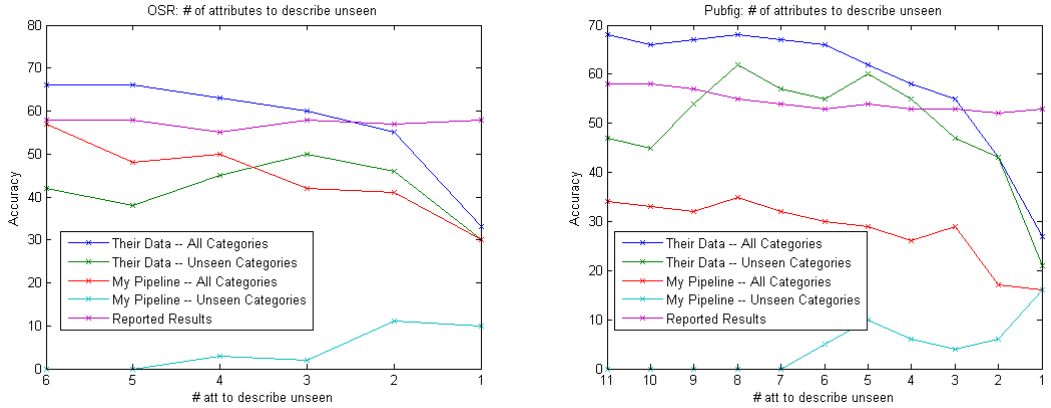

Figure 3. Replication results of Figure 1.5

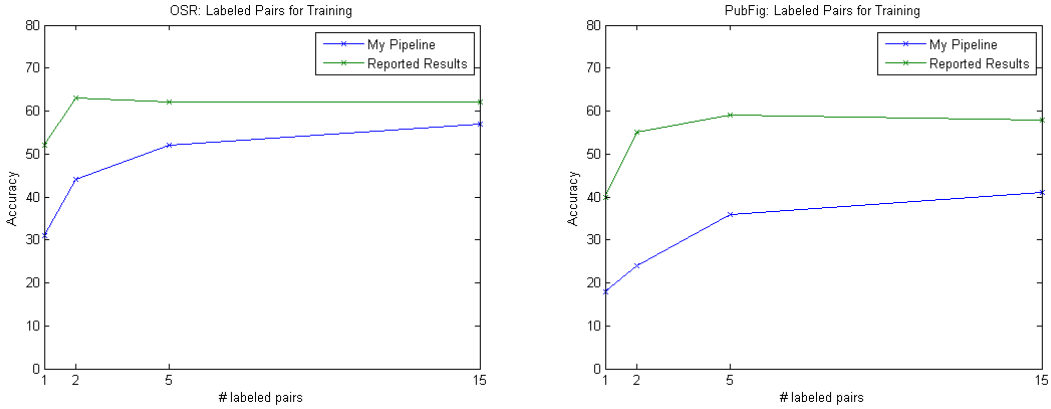

Figure 4. Replication results of Figure 1.6

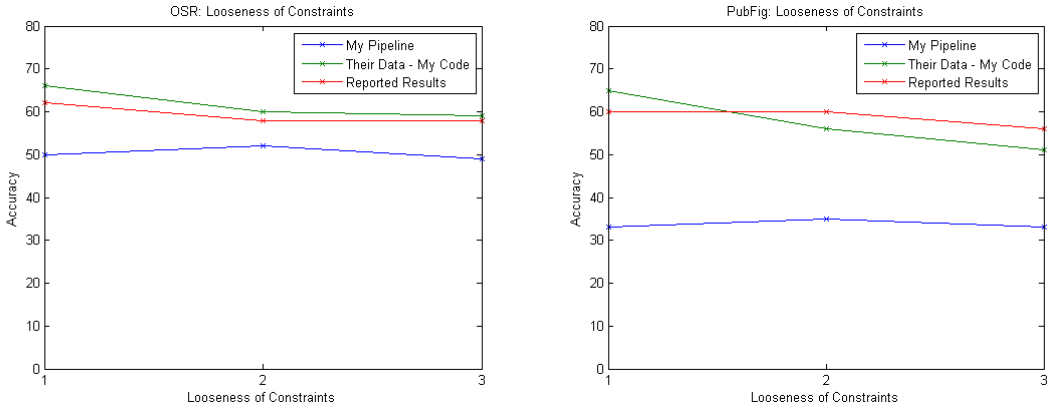

Figure 5. Replication results of Figure 1.6

Replication Procedure:

I began to perform this replication work by first accessing the data that they utilized to perform their experiments. The data utilized was the complete Outdoor Scene Recognition dataset, and a subset of the PubFig dataset. This data is available at the paper webpage at http://filebox.ece.vt.edu/~parikh/relative.html. The authors' have also posted the code that they used to train the ranking algorithm, and I downloaded this so that this could be used to check my ranking function results. The final piece of code that needed to be collected was the implementation of the GIST feature extractor, which was used to extract features from the candidate images, and the inventor of this feature, Antonio Torralba, makes this code feely available on his website (http://people.csail.mit.edu/torralba/code/spatialenvelope/).

I then began to extract features from the images that would be used for this task. I extracted GIST features from both the OSR and PubFig. Gist features are a 512 dimensional float vector. For the PubFig images I also converted the image into lab color space and extracted a 30-dimensional color histogram for this space. I then concatenated this vector with the GIST descriptor to create a 542 dimensional vector to describe the images. The paper states that the author's used a 45 dimensional histogram of lab features, but the extracted features provided on the paper's website use a 30-dimensional vector, so I chose to use this. Once the features were extracted I compared the features I collected to the features provided with the data from the author, and although they were not exactly the same they were very close to each other.

The authors provided in a .mat file the features, learned weights from their ranking implementation, and then which images were used for training. I then augmented the matlab implementation of RankSVM, created by Olivier Chapelle that can be found at http://olivier.chapelle.cc/primal/ranksvm.m, so that it worked with the ranking formulation proposed in the paper. This is the same procedure that the author's used. Once I had completed this work I then tested the output of the ranking algorithm, comparing the author's version to mine. The results were identical to each other and therefore I know that I have implemented this portion of the pipeline accurately.

After completing the ranking function I then implemented the generative probability model described in Section 3.2 of the paper. Throughout this work I attempted to create a pipeline that would allow for automated testing of the relative attributes work given any of the parameters that were augmented within the paper. Therefore, I created a Matlab matrix that held the order of each classes rank with respect to every other class for each attribute. I then carried

out the learning framework that was described within the paper with a slight augmentation. If an unseen class had the same attribute rank as a seen class for a particular attribute I assigned the mean rank for that attribute of the unseen class to the mean rank of that attribute for the seen class. Doing this slightly improved the results that I was receiving across the pipeline. Other than this slight change I followed the description in Section 3.2, and believe my pipeline to have replicated the author's work, because I was able to very closely replicate their results on the data set that they provided with my generative learning framework.

Finally, having finished these sections I began the final implementation portion of this work which included automatically creating the class training pairings based on the randomly chosen seen and unseen classes, number of labeled pairs, and number of attributes used for classification. This was the most difficult portion of my project, and to the best of my knowledge has not been completed correctly. The results in Figures 2-5 show that using the actual learned weights given by the authors' with my generative learning framework produces results that are very similar to those provided by the authors' in the paper. However, using my ranking algorithm I found that the result that I get in classification, especially on unseen classes is very poor in comparison to their results. Due to the fact that I have tested the output of their ranking function and mine together and they perform similarly I must deduce that I am selecting the training pairs in a different way than the author's. Below is an excerpt from the paper as to how the training pairs were collected for the experiments.

> *"Unless specified, we use 2 unseen and 6 seen categories. To train the ranking functions, we use 4 category pairs among seen categories, and unseen categories are described relative to the two closest seen categories for each attribute (one stronger, one weaker). We use 30 training images per class, and the rest for testing"*

I assumed the above excerpt to mean that for each testing iteration 4 class pairs were selected and each attribute was trained with relative pairings generated from these class pairs. All 30 training images from both pairs were used as inputs to my ranking function. There are multiple possible problems within this step that could be causing my results to be different than their results. The first possibility is that I am training differently than the authors' of the paper. I have three different training schemes implemented but none of them appears to be the same scheme that was used by the authors'. The other possibility is that my implementation of the generative learning framework is wrong in some way, and this is causing the poor performance particularly in the unseen categories. It can be seen in Figure 3, that for the unseen categories I am correctly classifying these images with 0% accuracy. This constitutes the numerical difference between my performance, and their performance. Therefore, it is likely that I have not been able to reproduce some portion of the learning framework. However, it can also be shown that by using their outputted ranks for each image my generative learning framework performs as well if not better than theirs due to the slight change in implementation described earlier. Therefore, there is also evidence to point to this not being the issue and that my underlying training scheme is wrong.

This has been a very enlightening experience, and I believe that 95% of the author's work has been reproduced accurately. However, one small issue is truly holding up the complete replication of the author's results. My next steps are to contact the author and ask them explicitly how they trained their ranking function, as well as continue to debug the training process to try to uncover any possible issues that are causing the drop in performance.