BRIDGING THE GAP: SIMULTANEOUS FINE TUNING FOR DATA RE-BALANCING

John McKay^{1,2}, Isaac Gerg², & Vishal Monga¹

Dept of Electrical Engineering & Computer Science¹, Applied Research Laboratory² Pennsylvania State University

ABSTRACT

There are many real-world classification problems wherein the issue of data imbalance (the case when a data set contains substantially more samples for one/many classes than the rest) is unavoidable. While under-sampling the problematic classes is a common solution, this is not a compelling option when the large data class is itself diverse and/or the limited data class is especially small. We suggest a strategy based on recent work concerning limited data problems which utilizes a supplemental set of images with similar properties to the limited data class to aid in the training of a neural network. We show results for our model against other typical methods on a real-world synthetic aperture sonar data set. Code can be found at github.com/JohnMcKay/dataImbalance.

Index Terms— Data Imbalance, Sonar Automatic Target Recognition, Neural Networks, Simultaneous Training

1. INTRODUCTION

The goal of any "re-balancing" scheme is to convince an algorithm to not disregard an underrepresented class. This is nontrivial as learned algorithms are incentivized to perform well on their training and if they see an overwhelming number of a certain class, they are going to be more apt to classify inputs in that direction. When it comes to sonar target recognition, it is obvious that such a tendency will be dangerous.

How do people typically handle data imbalances? For natural, optical images problems it is common to under-sample the large class [1]. This means purposefully removing training samples when training, say, a neural network. This and variants that compensate using synthesized data may work well in certain cases [2], but when it comes to sonar (or radar), omitting background elements that could contain unique debris or distinctive rocky patches is heading towards a direction of *less* information making it to the model. This can lead to confusion later on when tested on field data.

In [3], the authors present a novel manner of training convolutional neural networks (CNNs) when dealing with small data sets. They suggest drawing images from a supplemental, larger (source) data set that shares low-level features with the target data and, when training a CNN, propose simultaneously learning shared initial layers with the target and source data sets. This helps alleviate over-fitting and allows the deeper, finer layers to extract more information. We see that such a strategy can be adapted for the data imbalance problem in that a supplemental collection of images from a source data set can be used to draw images that are similar with respect to low level features to the data limited class and are dissimilar to the larger class. This discriminative parsing of the supplemental data set corrects for the data imbalance while not sacrificing information pertaining to the larger class.

In the following, we look to: formulate and detail our novel discriminative adaptation of [3] for data imbalances and demonstrate its potential with an undersea identification problem using real synthetic aperture sonar (SAS) images. Section 2 goes through how images from a supplemental data set are chosen. Section 3 goes through our CNN architecture and how the simultaneous training works. Lastly, Section 4 contains experimental work using the aforementioned actual SAS data. Note, we use *target data set* (D_t) to refer to the entire collection of images we want to classify, *source data set* (D_s) to refer the set of images we are drawing from to supplement our target data set, *starved class* (c_s) to refer to the limited data class of D_t , and *large class* (c_ℓ) to refer to the larger data class of D_t .

2. SUPPLEMENTAL DATA SELECTION

Our scheme for "re-balancing" data depends highly on picking the "right" images from D_s . That is, we need to select images from D_s that are similar to c_s and dissimilar to c_ℓ all according to some metric that detects meaningful characteristics. Let's start with the metric aspect; we know that trained convolutional neural networks have initial layers that resemble Gabor filters [4]. These edge filters are then followed in networks by layers that extract finer and finer details [5]. For our purposes, the later layers are not of much interest; we do not expect images in, say, Imagenet [6] to share high level characteristics with lobster crates under the sea. That said, we can expect that some images to share similar low-level edge characteristics. This is the crux of the idea behind using histograms of Gabor filter activations as the feature vectors for deciding "nearness." For every image in D_t and D_s we

Funding provided by ONR N00014-15-1-2042

Acknowledgements to Tiantong Guo¹, Tiep Vu Huu¹ for code advice



use Gabor filters to get edge maps and then get histograms of their intensity values. These activation histograms are then concatenated according to their image and this vector serves as our feature transformation.

This is analogous in some ways to what [3] did with limited data. They used the filters of an existing, trained network (Alexnet [7]) in conjunction with Gabor filters. These additional filters were only marginally helpful for our purposes and not worth the larger vector and increased computational stress. When dealing with more intricate images than our SAS example (i.e. ones that have color, have higher resolution, etc) such an action may be warranted.

With these feature vectors, the next step is to decipher which are close to c_s and far from c_ℓ to which we employ a nearest/farthest neighbor approach. We go through the edge features according to every source image and find their distance to the members of D_t . We then parse this data in two steps: first pluck out the N closest source images to a member of c_s and then refine that list to the M < N farthest from the members of c_ℓ . The idea is that we first want to ensure that our pool of potential supplemental images are, foremost, similar to the limited class and then we can make selections based on the distances away from the large class. This two stage approach does not require any additional calculations beyond a single nearest neighbor implementation, but this is still a nontrivial computational expense. We suggest using a random sample $\hat{D}_s \subset D_s$ to keep computations manageable.

Overall, the idea is this: for each $y \in D_t$, obtain their histogram features according to filter $f^{(i)}$, $h_y^{(i)}$ and concatenate each one over i = 1, ..., F histograms into a single vector $h_y = [h_y^{(1)}, ..., h_y^{(2)}]^T$. Do the same for the images in the source data set, $x \in D_x$, to obtain h_x and then calculate the distance between each of the source and target histograms, arriving in a scenario shown in Figure 1 (We suggest the L2 norm). The final selection is then done in two parts: isolate the N source images that have the smallest distance between them and a member of c_s and then, of that set, sort them by their nearest distances to a member of c_ℓ and keep the M farthest. Both parts of the final step are described in Figures 2 and 3. We let $U \subset \hat{D}_s$ be the set of images from \hat{D}_s that have been chosen to supplement class c_s .

Note that, in some ways, we are designing a scheme similar to the well-known SMOTE method used for support vector machine and similar classifiers [8, 9]. Instead of crafting synthetic samples, our use of existing data circumvents a generation step. This means we in principal are pursuing the same idea as SMOTE (obtaining representative features for learning) with a clever work around tailored for neural networks.

3. SIMULTANEOUSLY TRAINED NETWORK

The reason we get U is so that we can use it for training. Before we go forward, it is worth mentioning the common practice of weight sharing and transfer learning. When dealing with limited data or initialization problems, there is an idea of using existing weights from heavily trained models like VG-Gnet [10], Alexnet, etc. where authors had ample resources to train their networks on millions of images. Since the differences between natural images are relatively small, models can be *finely tuned* by starting with those existing weights and then trained from there with the target data set [5].

Simultaneously trained models (STMs) are similar in concept but differ in implementation. STMs start from randomized weights and flip between batches of U and D_s . This means, instead of crafting a model using the source data set and then imposing new information via fine tuning with D_t , STMs start anew and have D_t and U struggle against one another during the entirety of training. As the only shared quality between D_t and U are the edge features, this keeps STMs from getting overly influenced by c_ℓ regardless of how long they are trained. This is crucial for our re-balancing scheme.

4. SONAR TARGET RECOGNITION

Sonar automatic target recognition (ATR) suffers from extreme data imbalances [11] and we designed an experiment to illustrate the potential of our method as a viable option in this domain. We looked to classify objects (lobster crates) from



Fig. 5. Lobster crates

(top four) and clutter (bottom four) images.

undersea clutter using real-world data. Our images came from a synthetic aperture sonar system equipped to an unmanned underwater vehicle that scoured along New England's coast. The entire data set consisted of approximately 1.71km^2 of underwater area coverage. 1.12km^2 was designated as training and supplied $169,413 \ 168 \times 168$ patches with 869 of those containing crates (which we upsampled by a factor of ten for data augmentation) and the rest clutter (a 194:1 backgroundto-crate ratio). 0.58km^2 was used as the testing set and gave 89,048 patches consisting of 757 with crates and the other 88,291 as clutters (a 117:1 background-to-crate ratio).

Evaluating the effectiveness of a classifier on imbalanced data is nuanced. Typically, one sees a ROC curve when dealing with a binary problem but they are ill-equipped for skewed data scenarios; ROC curves *under emphasize* the effect of large numbers of false positives [12]. Instead, we looked at two more informative metrics: precision-recall (PR) and false alarm rate (FAR) curves. PR curves are less biased than ROC curves as they do not consider true negatives which overwhelm ROC curve statistics [12]. FAR curves replace a ROC curve's false positive rate with the expected number of false alarms per square 0-1km (i.e. a hard cut-off at 1km²) and are standard for ATR problems [13].

We looked to use the Caltech 256 data set [14] as a D_s and built a STM with N=18,000 and M=12,000. For context, we also built: a CNN with no source data influence and trained with a subsampling of the larger class, a CNN with class weights trained on an imbalanced data set, a CNN that was fine tuned using D_s and then trained on the imbalanced set, a CNN that was fine tuned with D_s and then trained using a subsampled data set, and, lastly, a simple nearest neighbor (i.e. non-CNN) scheme that used a distance-based scoring metric so we could illustrate its PR/FAR performance. Each competing scheme offers a different view of against our STM model. Note for the CNNs, we used evaluated the AUC of a PR curve on a small validation set at each epoch and chose the weights based on the optimal epoch. The fine tuned models were pre-trained on D_s for 30 epochs based on over-fitting and general performance.

In our experiments, the STM outperformed the five others in testing and, as Figures 7 and 8 show, there was a considerable gap. Our results revealed a trend: models that could use the full, imbalanced training did better than subsampled ones. Even the nearest neighbor scheme with the full training set did better in terms of FAR than the two subsampled cases, reflecting the benefit of more information. The STM's relative success unveils the power in our discriminative, simultaneous training scheme; if D_s were to be used for fine tuning without any refinement, it is arguably as powerful as just weights.

We lastly note an interesting phenomenon with regards to the training loss. As shown in Figure 6, the D_s -using models asymptoted to zero yet the fine tuned with subsampling failed to achieve the quality of the others, suggesting a different local minima convergence. In later work, it would be prudent to investigate the loss function manifold to understand how our and other training schemes impact its geometry.

5. CONCLUSION

We have shown that STMs using a discriminatively chosen source set U can help alleviate the problematic trade-off between incorporating more large-class information into a



model and the bias that causes. A further investigation into the choice of D_s or edge-feature statistics may be a fruitful direction for future research, but for now we consider our work a compelling option for those struggling with imbalanced training sets, especially in the case of sonar ATR.

6. REFERENCES

- H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data En*gineering, vol. 21, no. 9, pp. 1263–1284, Sept 2009.
- [2] S. Barua, M. M. Islam, X. Yao, and K. Murase, "Mwmote–majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, Feb 2014.
- [3] Weifeng Ge and Yizhou Yu, "Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [4] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, "Understanding neural networks through deep visualization," arXiv preprint arXiv:1506.06579, 2015.
- [5] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.
- [6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

- [7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information* processing systems, 2012, pp. 1097–1105.
- [8] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelli*gence research, vol. 16, pp. 321–357, 2002.
- [9] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz, "Applying support vector machines to imbalanced datasets," *Machine learning: ECML 2004*, pp. 39–50, 2004.
- [10] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [11] Jason Stack, "Automation for underwater mine recognition: current trends and future strategy," in SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, 2011, pp. 80170K–80170K.
- [12] Jesse Davis and Mark Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings* of the 23rd international conference on Machine learning. ACM, 2006, pp. 233–240.
- [13] Timothy D Ross, Steven W Worrell, Vincent J Velten, John C Mossing, and Michael Lee Bryant, "Standard sar atr evaluation experiments using the mstar public release data set," in *Algorithms for Synthetic Aperture Radar Imagery V*. International Society for Optics and Photonics, 1998, vol. 3370, pp. 566–574.
- [14] Gregory Griffin, Alex Holub, and Pietro Perona, "Caltech-256 object category dataset," 2007.