

Understanding classifier errors by examining influential neighbors Supplementary Material

1 Measuring the influence of a training example on a test example

As described in Section 5.1, we compared how well a suite of distance metrics could determine if a training example had influence on a given test example, where influence is defined as in Equation 1. This requires computing the “groundtruth”, according to Equation 1, of whether a training example has influence on a test example for many pairs of training and test examples. Computing influence as defined is computationally prohibitive, as it requires taking the expected value over all reasonable classifiers given two different data sets. We thus made several approximations when computing this groundtruth.

First, instead of using the expected difference in binary predictions over classifiers in the version space, we used the differences in the continuous prediction scores of two classifiers:

$$J(x' \rightarrow x) = f_1(x) - f_0(x) \quad (1)$$

where f_1 and f_0 are the boosting classifiers trained from $\mathcal{D}_1 = \mathcal{D} \cup \{(x', 1)\}$ and $\mathcal{D}_0 = \mathcal{D} \cup \{(x', 0)\}$, respectively. We use the continuous scores as they are a proxy for the classifier’s confidence.

To distinguish changes in predictions on a test example due to noise from true influence, we trained 10 classifiers from each training data set \mathcal{D}_1 and \mathcal{D}_0 with some randomness introduced to the training procedure: instead of finding the best rule using all the examples and their weights at each iteration, we found the best rule by sampling the training examples according to the weights assigned to them in that iteration. We labeled that a training example has influence on a test example if its score changed by more than twice the combined standard deviations:

$$\frac{\mu_1(x) - \mu_0(x)}{\sqrt{\sigma_1(x)^2 + \sigma_0(x)^2}} > 2 \quad (2)$$

where $\mu_1(x)$, $\sigma_1(x)$, $\mu_0(x)$, and $\sigma_0(x)$ are the means and variance of the predictions on x for the 10 classifiers trained using \mathcal{D}_1 and \mathcal{D}_0 , respectively. These

modifications ensured that pairs of examples tagged as having influence truly had influence (i.e. minimized the number of false positives from noise).

Finally to again separate true influence from noise, we magnified the effect by increasing its weight to 15% of the total weight for that class.

2 Similar Examples with inconsistent labels from the behavior data set

Figure 1 shows examples from behavior data set that are visually similar but had opposite labels. These examples were automatically found using the dissimilarity metric by search for examples that were close but had opposite labels in the training data set.

3 Application to the ImageNet data set

We used with the DeCAF₆ [1] representation of each image – the activations of the 6th hidden layer of the deep convolutional network. This was input into a variant of the GentleBoost algorithm, as described in [2]. For each of the 28-categories, a one-vs-all classifier was trained. We weighted errors on each example such that the total weight for positives and negatives was equal. For speed of learning, the number of samples considered at each level of boosting was 2,500, and each feature was binned into 100 bins when searching for the optimal decision stump threshold. A multi-class classifier was obtained by predicting the class whose corresponding classifier had the highest output score.

References

- [1] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.
- [2] Mayank Kabra, Alice A Robie, Marta Rivera-Alba, Steven Branson, and Kristin Branson. JAABA: Interactive Machine Learning for Automatic Annotation of Animal Behavior Detection. *Nature Methods*, 10(1), 2013.



Figure 1: Similar examples with inconsistent labels that were found in the behavior dataset by searching for examples that are close but have opposite labels. In each case, the center frame had opposite labels even though the trajectories are almost identical.