Bootstrapping Image Classification with Sample Evaluation

Venkatraman Narayanan  
The Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
venkatraman@cmu.edu

Ruta Desai  
The Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
rutad@cmu.edu

Sanjiban Choudhury  
The Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213  
sanjiban@cmu.edu

Abstract

In this work, we look at the problem of multi-class image classification in a semi-supervised learning framework. Given a small set of labeled images, and a much larger set of unlabeled images, we propose a semi-supervised learning method that combines bootstrapping with sample evaluation, to continuously update the learned models for each class. Bootstrapping involves using self-labeled images to re-train the learned models. To overcome the semantic drift that naive bootstrapping is prone to, we use additional sample evaluation methods based on the ideas of co-training and pairwise constraints, to determine whether or not a newly classified instance should be used for re-training. Experimental results show the usefulness of sample evaluation, when used in conjunction with bootstrapping. In particular, our method is able to achieve a 8% improvement in overall accuracy over baseline bootstrapping, on a 15 class subset of the SUN (Scene UNderstanding) dataset.

1 Introduction

Image classification using supervised approaches is limited by the availability of annotated images. Given the large number of unlabeled images available on the web, it is natural to think of a semi-supervised learning approach, where only a few labeled instances are required. For instance, it would be straightforward to obtain unlabeled images by crawling image hosting websites, whereas it is much more expensive to annotate all those images. A commonly used semi-supervised learning approach is bootstrapping, which starts off with a small number of labeled samples and iteratively re-trains the classifiers based on self-labeled images. Specifically, an initial model which is learnt from a set of labeled seed examples is used to label a portion of the unlabeled data. The initial model is then re-trained on the combined set of seed examples and the self-labeled examples. This process is repeated over multiple batches of unlabeled data.

Bootstrapping in its vanilla form is prone to semantic drift because the self-labeled instances deviate significantly from the seed examples over time, owing to accumulated errors. As an example, a classifier that has initially learnt that all dogs are cats, will continue to classify dogs as cats, and use them as positive training instances for the cat category. This creates a positive feedback cycle, that causes the learner to rapidly latch on to a concept that has been learnt wrongly. The focus of this
work is to refine the bootstrapping process by promoting only those examples that would improve the learner's performance when trained upon them. To this effect, we propose a set of sample evaluation techniques that help determine the set of examples to be used for re-training. These techniques will be discussed in Sec. 3.

2 Related Work

In the domains of natural language processing and information extraction, various semi-supervised learning approaches based on bootstrapping have been proposed for webpage classification, parsing and named entity classification [2]. These approaches employ contextual patterns and relations among text data to mitigate semantic drift. To constrain the semi-supervised learning process, some approaches [2, 4] also use the **mutual exclusion** principle, wherein positive examples of one class automatically become negative examples for all other classes. However, there has not been a clear extension of mutual exclusion and coupled learning to the domain of image classification.

Co-training [1] has also been explored for exploiting conditionally independent features to perform type-checking before promoting new instances during bootstrapping. In co-training, two different learners are trained using separate views or features of seed examples, and their performance on the unlabeled examples determines which instances are promoted for bootstrapping. It has been shown [1] that using multiple views of data decreases the number of unlabeled examples that are required to learn classifiers with arbitrarily high accuracy. In our work, we draw upon the idea of co-training to constrain the examples that are used for bootstrapping.

Recently, Shrivastava et al. [6] proposed an analogous framework for image classification, by manually defining comparative relationships between different image classes to constrain bootstrapping. However the origin of these relationships are semantic in nature, in addition to being defined by a human annotator. This limits the scalability of the approach and introduces semantic definitions, which might be hard to learn from data. In this work, we instead focus only on visual features, to avoid problems arising from semantic constraints.

3 Method Overview

In this section, we provide a formal definition of the problem we are trying to solve, and discuss the technical details of our method. Given a set of labeled images \( \mathcal{L} = \{X, Y\}, Y \in \{1, 2, \ldots, N\} \) and unlabeled images \( \mathcal{U} = \{X\} \), the objective is to learn a target function \( f : X \rightarrow Y \) that minimizes the true risk \( P(f(X) \neq f^*(X)) \) over the actual distribution of data, where \( f^* \) is the true target function. We tackle this problem using the well established idea of bootstrapping, wherein an initial target function \( f_0 \) is learnt using only images in \( \mathcal{L} \) and improved iteratively, using self-labeled images from \( \mathcal{U} \). However, as mentioned earlier, this naive bootstrapping scheme is susceptible to semantic drift and the target function \( f \) that is learnt rapidly deviates from \( f^* \). To counter semantic drift, we propose three sample evaluation techniques that constrain the use of unlabeled data for re-training. We discuss the sample evaluation techniques in the sections that follow. Figure 1 provides an overview of this bootstrapping pipeline.

3.1 Suppressing non-informative samples

A key assumption that we make here is that the target function \( f \) learnt from the seed data is a reasonable approximation of \( f^* \). We can then say that images from \( \mathcal{U} \) that are classified with a high confidence would not be very informative during the re-training process. The idea is similar to the concept of mining for hard negatives, except that we do not know the true labels here. Formally, given a sample \( u \in \mathcal{U} \), we declare \( u \) as an informative sample only if \( P(f(u) = Y) < P_{\text{thresh}} \forall Y \in \{1, 2, \ldots, N\} \). For implementation purposes, we approximate \( P(f(u) = Y) \) from the scores returned by a one-versus-all SVM classifier. If we assume an SVM framework, yet another intuitive explanation for this technique is that we discard samples far away from the margins, as they do not influence the margin at all. We also note that doing the reverse, i.e., promoting only the ‘confident’ samples, results in poorer performance, which is counter-intuitive. One possible explanation is that there is not much to be gained by training on an instance which could already be labeled confidently.
However, in practice, the assumption that the hypothesis learnt from the seed data closely approximates the true hypothesis does not hold when the number of seed examples used is very small. This will be elaborated upon in Sec. 4.

### 3.2 Independent evaluation

A major cause of semantic drift is the positive feedback effect that arises from the learner re-training on self-labeled examples. In an ideal scenario, an oracle would give us the labels for the unlabeled images, which would then solve the self-labeling issue. In practice, one way to simulate an oracle would be to use another learner, that has been trained independent of the original learner. This
train of thought leads to the well known co-training algorithm, where two learners that are trained independently on two views of the data help each other through the bootstrapping process. However, we differ from the co-training algorithm in that we only promote instances on which both the learners agree, as opposed to independently promoting the most confident positive and negative examples returned by each of the learners. Formally, given two learned target functions \( f_1 \) and \( f_2 \), an instance \( u \in \mathcal{U} \) is promoted as a training sample only if \( f_1(u) = f_2(u) \). The promoted instances are added to the pool of labeled images which are then used to re-train the main learner, as well as the independent evaluator.

3.3 Enforcing discriminative constraints

Classes with similar visual features are highly prone to be confused with each other during classification. Promotion of such classes further contributes to semantic drift. The idea behind enforcing discriminative constraints is based on the fact that the confusion matrix from a validation dataset is a good indicator of easily confusable classes. We use the information from the confusion matrix (computed on a validation set), to train individual one-versus-one SVM classifiers that can be used to discriminate between pairs of confusable classes. Formally, two pairs of classes \( i \) and \( j \) are likely to have been confused if \( C(i, j) > C_{\text{thresh}} \) or \( C(j, i) > C_{\text{thresh}} \), \( i \neq j \), \( \forall i, j \in \{1, 2, \ldots, N\} \). Given the target function \( f \) and the discriminative classifier \( g_{i,j} \), an instance \( u \in \mathcal{U} \) is promoted as a training sample only if \( f(u) = g_{i,j}(u) \), \( f(u) \in \{i, j\} \). The threshold \( C_{\text{thresh}} \) is a free parameter—smaller thresholds will identify more confused class, at the cost of training more pairwise discriminative classifiers. Figure 2 illustrates this process of obtaining pairs of confusable classes, and the corresponding one-versus-one SVM classifiers.

The independent evaluators and one-versus-one SVM classifiers allow us to constrain bootstrapping using only visual features, thereby eliminating the need for semantic constraints which might need human intervention as in [6].

4 Experiments

In this section, we provide details about the images, categories, features and classifiers. For our baseline, we use naive bootstrapping, with no sample evaluation at all. That is, we iteratively promote batches of self-labeled data for the training phase, without any constraints on the self-labeled examples. Results of bootstrapping with different sample evaluation techniques are compared against the baseline.

4.1 Database

We use 15 natural scene categories dataset from the UIUC [5] and SUN database [7] containing a total of 8975 images for our results. The categories are - house, coast, forest, highway, airport terminal, mountain, golf course, street, tall building, conference room, bedroom, church, kitchen, living room and market. Each category has 300 to 800 images, and average image size is 300 × 250 pixels. The categories range across indoor, natural and man-made scenes. In our experiment, we use 5 seed examples from each category for training, 30 examples for validation, 30 examples for final testing and the remaining as unlabeled examples. The unlabeled examples are handled in a batch size of 1000. These set of images provide grounds for comparison between the different proposed methods.

4.2 Features

To determine the most suitable feature or set of features for classification, we considered a selection of several state-of-art features using the publicly available code from [7]. These features were then used to train one-versus-all support vector machines, and their performance is shown in Fig. 3. The best performing feature was Histogram of Oriented Gradients (HOG), which are densely extracted on a regular grid at steps of 8 pixels. The performance of the HOG features can be attributed to their ability to capture the nature and density of gradients in an image. Another feature of noteworthy performance was the texton histogram, which essentially clusters responses to a filter bank of 8 orientations, 2 scales and 2 elongations. Since this feature also captures low frequency components
of the image, it is conditionally independent of the HOG descriptor, given the image class. Thus, we picked the HOG feature for our main classifier, and the texton histogram for training the independent evaluator classifier.

4.3 Classifiers

The classifier used is the one-versus-all support vector machine (SVM). Although this type of SVM provides an independent decision boundary for each class and could lead to uncalibrated scores, it has shown to be successful in practice [7]. The classifier used for the discriminative evaluator is a one-versus-one SVM trained specifically on two classes. Histogram intersection kernel [5] is used to compute similarity between histograms of oriented gradients, whereas distances between texton histograms are computed using normalized $\chi^2$ distance. LIBSVM [3] was used to compute the support vectors.

4.4 Results

The performance of all methods over the test data with increasing number of iterations averaged over 10 separate trials are shown in Fig. 4. The baseline bootstrapping accuracy is 56.58%. There is a distinct improvement in performance (60.16%) after the introduction of the independent evaluator. In this case, the curve jumps at the second iteration and continues to rise steadily. This is indicative of the independent evaluator suppressing images which are misclassified. The monotonic nature of the curve shows that since both the classifiers are learning, semantic drift is arrested. The discriminative classifier also shows a performance increase to 59.91%. This improvement is due to resolving classes that are clearly confused, for example, living room and kitchen. On combining the discriminative classifier and independent evaluator, the performance dramatically increased to 64.17%. Moreover, the curve continues to rise instead of flattening out. This is because the two evaluators work in tandem, one improving the nominal performance of the system while the other targeting specific confusion. It was interesting to observe that promoting hard negative alone performed poorer than baseline. This indicates that due to the small number of seed samples, the classifier did not learn the nominal examples well enough.

The confusion matrix after incorporation of the discriminative classifier is show in Fig. 5. In this particular example, class 13 and 14 (kitchen and living room) are particularly confused as indicated by the magnitude of the off-diagonal entries. This is considerably reduced at the end of 8 iterations and both classes have comparable accuracy.
Figure 4: Plot of overall accuracy for 15 classes, with increasing number of iterations, for different sample evaluation methods. Accuracies are averaged over 10 trials with random seed data initialization.

Figure 5: Discriminative classifiers resolve confused classes. Left: Confusion matrix before bootstrapping. Right: Confusion matrix after bootstrapping.
In Fig. 6, we take a look at the type of images for the class living room being promoted by different evaluation methods. The baseline approach promotes examples which are incorrect, thus indicative of semantic drift. The independent evaluator promotes better examples, but cannot overcome the particularly confusing examples. The discriminative classifier overcomes the difficult case, but suffers for not incorporating more than one view. However, when they are used together, their performance is far superior.

5 Conclusions and Discussion

We described a semi-supervised learning framework for multi-class image classification, based on constrained bootstrapping. The major contributions of our work are the introduction of sample evaluation techniques based on an independent evaluator and pairwise discriminative classifiers. The independent evaluator exploits multiple views of the data, while the pairwise discriminative classifier resolves ambiguity between highly confusable classes. Together, the two techniques are able to mitigate semantic drift during bootstrapping. Moreover, by learning sample evaluators from data, we eliminate the need for manual definition of constraints. Our experiments on a subset of the SUN (Scene UNderstanding) dataset show that our method is able to provide a 8% boost over baseline bootstrapping.

On the flipside, certain parameters of the framework have to be set based on computational resources. For instance, we could train several pairwise discriminative classifiers to resolve confusion between pairs of classes, while paying computational resource for training all the classifiers. Also, the performance of the system is affected to some extent by the batch size of the unlabeled data used at every iteration of bootstrapping. Smaller batch sizes lead to frequent re-training, which may in turn cause semantic drift. On the other hand, larger batch sizes take longer to re-train the classifiers. One solution to overcome the batch size issue would be to adopt an online learning approach, that updates the classifiers on the go.

References


