Table 1: Comparison between existing methods and our method

	Visual 20 questions game (Branson et al. 2010)	Flock (Cheng and Bernstein 2015)	Adaptive triple selection (Zou, Chaudhuri, and Kalai 2015)	**AdaFlock**
Crowdsourced feature-definition		✓	✓	✓
Crowdsourced feature-labeling	✓	✓	✓	✓
Adaptive example selection			✓	✓
Supervised learning		✓		✓

ples. Zou et al. proposed an adaptive method for selecting triples of examples to avoid the generation of overlapping features (Zou, Chaudhuri, and Kalai 2015). This method chooses examples based on feature labels of the previously discovered features. Because the method does not focus on supervised learning, the performance of a classifier is not considered during example selection.

In this paper, we propose a new algorithm called *AdaFlock* to efficiently generate informative features through crowdsourcing. Our algorithm aims to obtain features helpful for improving the classification performance through iterations. AdaFlock is inspired by AdaBoost (Freund and Schapire 1997), which iteratively trains weak classifiers by increasing the weights of examples misclassified by the current classifiers. Analogously, at each iteration of AdaFlock, crowdsourcing workers are shown examples selected according to the classification errors of the current classifiers (Figure 1(a)). The workers are asked to generate features helpful for correctly classifying the given examples (Figure 1(b)). AdaFlock then asks crowdsourcing workers to label each example based on each feature definition (Figure 1(c)). A weak classifier is trained by using the obtained labels, and the Filter function (Bradley and Schapire 2007) is applied for resampling examples according to the classification errors of the current classifier (Figure 1(d)). The sampled examples are shown to workers of the next iteration.

Table 1 summarizes the difference between AdaFlock and the other crowdsourced feature discovery methods. Although the method proposed by Branson et al. (2010) requires predefined feature definitions, AdaFlock uses crowdsourcing for defining features. In contrast to Flock (Cheng and Bernstein 2015), AdaFlock adaptively generates features to improve the classification accuracy. While adaptive triple selection (Zou, Chaudhuri, and Kalai 2015) aims to obtain diverse features, the goal of AdaFlock is to obtain informative features to improve the classification accuracy.

We conducted experiments on a crowdsourcing platform by using image and movie classification datasets and observed that AdaFlock discovers various types of features not covered by Flock. Moreover, a classifier built by using AdaFlock outperforms the classifier built through Flock.

It would be worth noticing that one of the major drawbacks of crowdsourced feature discovery is scalability; this approach is not suitable for classifying a large number of examples because of the requirement of crowdsourced feature labeling. Practical situations where crowdsourced feature discovery is useful are (1) obtained interpretable features themselves are important for explaining predictions, and (2) it is desired to create an accurate classifier regardless of crowdsourcing costs.

The contributions of this paper are threefold:

- We address the problem of adaptive crowdsourced feature generation for supervised learning.
- We propose a novel algorithm called AdaFlock, which obtains informative features by sampling difficult examples for the current classifier.
- Through experiments conducted using actual crowdsourcing datasets, we confirm that AdaFlock achieves better classification performance than the existing methods.

2 AdaFlock

2.1 Problem Setting

We first show a formulation of our feature generation problem for predictive modeling. We focus on binary classification in this paper. Assume that there is a training dataset as $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $y_i \in \{-1, +1\}$. Given the training dataset, our goal is to build a classifier $H(x)$. Unlike typical binary classification problems, a feature vector of each x_i is not given here. Thus, we will concurrently generate feature vectors and train a classifier.

2.2 Overview

We propose a novel algorithm called AdaFlock to efficiently generate feature vectors through crowdsourcing and to train a classifier by using the obtained features. AdaFlock requests crowdsourcing workers to process *feature-definition* tasks and *feature-labeling* tasks. Examples of these two tasks are illustrated in Figure 3 and 4. In feature-definition tasks, workers are shown a small number of positive and negative examples and asked to describe the difference between the positive and negative examples. This approach is called analogical encoding, and its advantages for producing predictive features have been demonstrated in (Cheng and Bernstein 2015). AdaFlock instructs workers to write a description as a yes–no question; for example, “is the sky illustrated in this painting?” or “does this text contain any hard number?” In feature-labeling tasks, workers are given