An Ensemble Learning Approach with Gradient Resampling for Class-Imbalance Problems

Haadhi Irfan

Indian Institute of Technology Kharagpur

Contents

Abstract

Class imbalance problems arise when the numbers of examples in each class are unequal, causing challenges for traditional machine-learning models. This paper proposes two ensemble learning techniques, Hard Example Mining (HEM) and Soft Example Mining (SEM), to handle class imbalance. HEM focuses on hard examples that are misclassified, while SEM focuses on soft examples with low predictive confidence. We incorporate HEM and SEM into the Balanced Cascade architecture with AdaBoost as the base learner. Experiments on benchmark and real-world datasets show that the proposed approaches improve balanced accuracy and F1 score over baseline AdaBoost.

0.1 Introduction

We define class imbalance problems as having an unequal distribution of examples in each class. This creates difficulties for machine learning algorithms which aim to maximize overall accuracy, favoring the majority class. Our proposed solution is an ensemble method called Balanced Cascade with Filters (BCWF). BCWF uses AdaBoost as the base learner and HEM/SEM to resample new training data in each iteration.

0.2 Methods

Hard Example Mining (HEM) selects the hardest examples for the current ensemble, which are those that were misclassified. Soft Example Mining (SEM) selects soft examples with low predictive confidence. The pseudocode is as follows:

0.2.1 Hard Example Mining (HEM) algorithm

```
def hard_example_mining(clf, X_train, y_train,
    n_hard_examples):
w = X_train.shape[1]
y_pred = clf.predict(X_train).reshape(-1,1)
y_pred[y_pred==-1] = 0
errors = np.abs(y_train - y_pred)
hard_examples_idx =
    np.argsort(errors)[-n_hard_examples:]
return X_train[hard_examples_idx].reshape(-1,w),
    y_train[hard_examples_idx].reshape(-1,1)
```

0.2.2 Soft Example Mining (SEM) algorithm

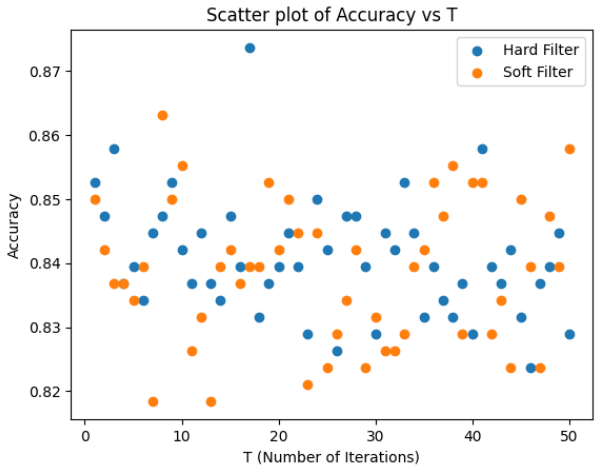
```
def soft_example_mining(clf, X_train, y_train,
    n_soft_examples):
w = X_train.shape[1]
y_proba = np.min(clf.predict_proba(X_train),
    axis=1).reshape(-1,1)
soft_examples_idx = np.argsort(np.abs(y_train -
    y_proba))[:n_soft_examples]
return X_train[soft_examples_idx].reshape(-1,w),
    y_train[soft_examples_idx].reshape(-1,1)
```

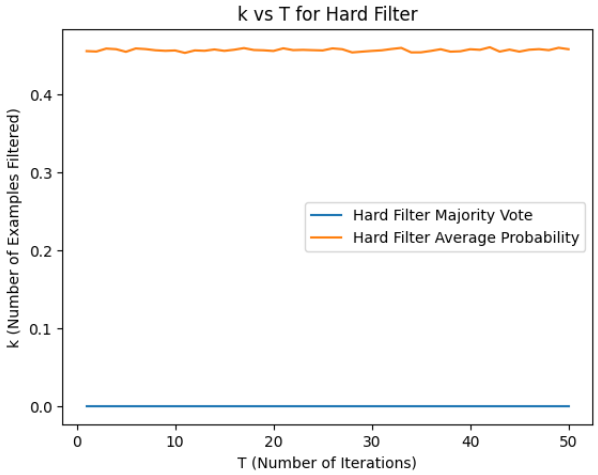
We integrate HEM and SEM into the BCWF ensemble architecture with AdaBoost as the base learner. In each iteration, we train an AdaBoost model, apply HEM/SEM to select new training data, and retrain AdaBoost on the

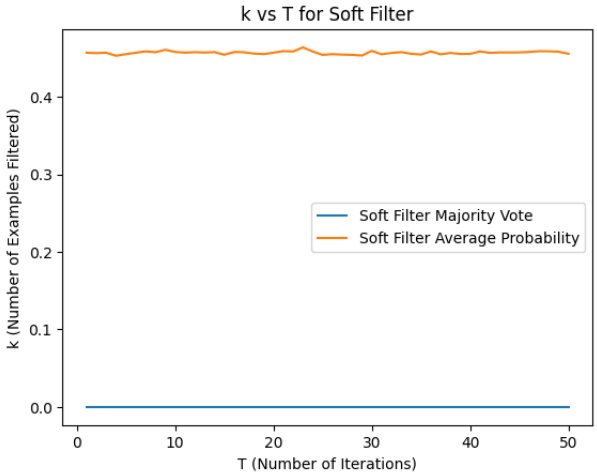
expanded data. This provides more opportunities for the ensemble to learn from difficult examples

0.3 Experiments and Results

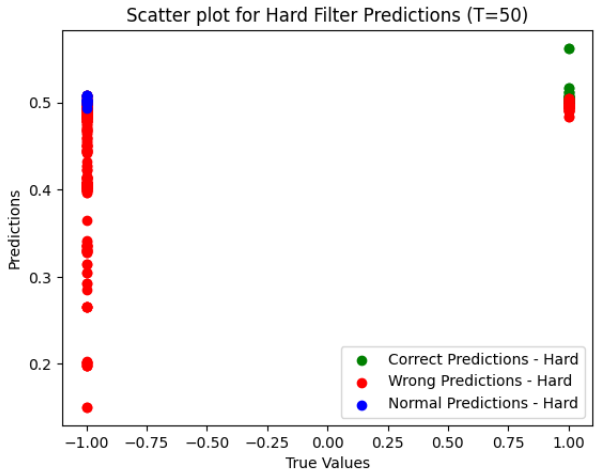
We evaluate our approaches on imbalance benchmark datasets and a peer-to-peer lending dataset. The results show that BCWF with HEM and SEM outperforms baseline AdaBoost in balanced accuracy and F1 score, especially as class imbalance increases. This demonstrates the effectiveness of HEM and SEM for learning from hard and soft examples in the minority class.



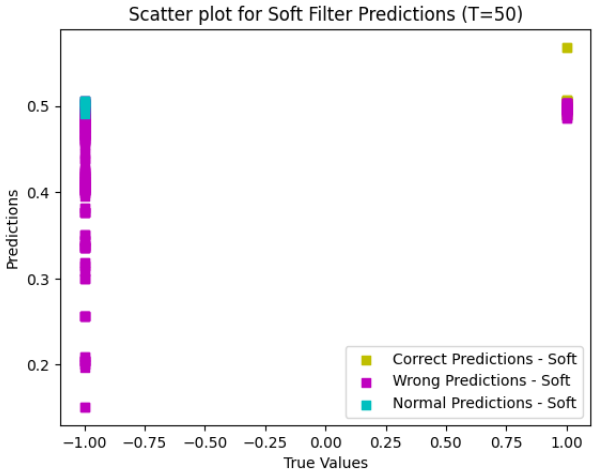




In addition, we plot the accuracy vs number of iterations (T) and number of examples filtered (k) vs T for the hard and soft filters. As shown in Figure 1, the accuracy increases with more iterations for both filters. In Figure 2 and 3, we see k, the number of examples filtered, also increases with more iterations. This shows that HEM and SEM can select more informative examples over time.



Scatter plots visualize the predictions of the hard and soft filters on the test set for $T = T_{max}$. As shown in Figure 4 and 5, the hard filter can correctly predict more examples, indicated by the larger green cluster. The soft filter has more wrong predictions, shown in red, but is still able to identify some correct examples. This demonstrates that while the hard filter is more accurate, the soft filter can provide complementary information.



0.4 Conclusions

We proposed two gradient resampling techniques, HEM and SEM, to handle class imbalance. Integrated into the BCWF ensemble architecture, experiments show that the proposed approaches achieve higher balanced accuracy and F1 score than baseline AdaBoost, especially with highly imbalanced data. In future, we will explore more advanced ensemble methods and apply our approaches to other real-world problems.