# IACAS at Web Spam Challenge 2007 Track I

Guang-Gang Geng, Chun-Heng Wang, Xiao-Bo Jin, Qiu-Dan Li, Lei Xu
Institute of Automation, Chinese Academy of Sciences
No.95, Zhongguancun East Road, Haidian District, Beijing P. R. China, 100080
{guanggang.geng,chunheng.wang,xiaobo.jin,qiudan.li,lei.xu}@ia.ac.cn

On the Web, reputable hosts are easily obtained, however collecting spam websites is relatively difficult. In reality, the ratio of spam sites are lower, which has the same situation in standard WEBSPAM-UK2006 benchmark. Our work focus on how to take full advantage of the information contained in reputable hosts. Based on the facts mentioned above, we treat web spam detection as a class-imbalance pattern recognition problem, and employ an ensemble classification strategy to train each classifier with randomly under-sampled samples, then aggregate them with accumulating the predicted spamicity.

Under-sampling has been popularly used in class-imbalance learning. Under-sampling uses only a subset of the major class examples to train the classifier, which was effective in many field. However, potentially useful information contained in the ignored examples, i.e. examples in $S \bigcap \overline{S'}$ are neglected, which is the main deficiency of under-sampling algorithm. In our work, we employ an ensemble strategy to overcome the deficiency and keep the efficiency of under-sampling.

In the ensemble method, we independently sample several subsets $S_1, S_2, ..., S_n$ from $S$. For each subset $S_i \quad (i \in N)$, a classifier $C_i$ is trained using $S_i$ and $M$. All the results generated by the sub classifiers are combined for the final decision. In web spam detection, the combination is based on the $PS(x, C_i)$, which is computed with formula $PS(x,C) = \frac{P_{spam}(x,C)}{P_{spam}(x,C)+P_{normal}(x,C)}$, where $x$ is a test sample, $C$ is a specific classifier, $P_{spam}(x,C)$ and $P_{normal}(x,C)$ is the classifier $C$ predicted probability of $x$ belonging to spam or not.

The implementation process is presented as follows:

1. Input a set of samples with minor class examples $M$ and major class examples $S$ ($M$ and $S$ corresponds to spam and normal set respectively), the times $n$ of resampling from $S$, and the sampling ratio $K$.

2. $i = 0$.

3. while $(i < n)$ { Randomly sample a subset $S_i$ $(|S_i| \leq |S|, \quad |S_i| = K * |M|)$ from $S$. Train $C_i$[1] with $S_i$ and $M$. Save the learned model $Model_i$. i=i+1. }

4. Input test sample $x$; for $(i = 0;\ i < n;\ i++)$ { Test $x$ with $Model_i$, and compute $PS(x, C_i)$ }.

5. $spamicity = 0$; for $(i = 0;\ i < n;\ i++)$ { $spamicity = spamicity + PS(x, C_i)$ }.

6. $spamicity = spamicity/n$; if $(spamicity >= 0.5)$ { $x$ is spam } else { $x$ is normal }.

In the algorithm, step 1–3 are the learning stage, and step 4–6 are the testing process. As all the sub classification process are independent, the algorithm is applicable to parallel computing.

The features used for classification involve transformed link-based features, content-based features and HostRank relevant features[2]. HostRank is similar to the original PageRank algorithm in spirit, where hosts are treated as the minimal granularity. Different weight strategy for host-level hyperlink are used, such as $1, N$ and $log(N)$, where $N$ is the link number between two hosts[3]. Based on the HostRank, features similar with transformed link-based features in form are extracted. The submitted result are computed with a combination of all the features without feature selection.

The ensemble under-sampling strategy is applied to bagging over C4.5 and adaboost with C4.5 as weak classifier, corresponding to the submitted ***IACASprediction1.txt*** and ***IACASprediction2.txt*** respectively. IACASprediction1.txt gives the predicted spamicity, and IACASprediction2.txt does not present the probability.

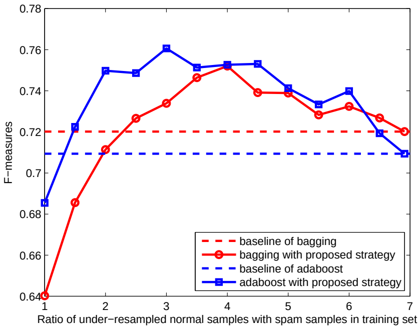

**Figure 1: Comparison of F-measure with Ensemble Under-Sampling Strategy**

Figure. 1 shows the comparison of F-measure with proposed strategy. The baselines were computed without under-resampling. The result were obtained using all of the labels set with 2 times 5-fold cross-validation, resampling times $n = 9$. Experimental results showed that the proposed learning strategy is robust, and could improve the web spam detection performance effectively.

[1] In the algorithm, $C_i$ may be C4.5, bagging or adaboost.

[2] The transformed link-based features and content-based features were obtained from the website of Web Spam Challenge 2007.

[3] The hyperlinks in the same host were not taken into account.

*Web Spam Challenge* '07, May 8, 2007 Banff, Alberta, Canada.
.