A Novel Field Learning Algorithm for Dual Imbalance Text Classification

Ling Zhuang, Honghua Dai, and Xiaoshu Hang

School of Information Technology, Deakin University,
221 Burwood Highway, VIC 3125, Australia
lzhu@deakin.edu.au hdai@deakin.edu.au xhan@deakin.edu.au

Abstract. Fish-net algorithm is a novel field learning algorithm which derives classification rules by looking at the range of values of each attribute instead of the individual point values. In this paper, we present a Feature Selection Fish-net learning algorithm to solve the Dual Imbalance problem on text classification. Dual imbalance includes the instance imbalance and feature imbalance. The instance imbalance is caused by the unevenly distributed classes and feature imbalance is due to the different document length. The proposed approach consists of two phases: (1) select a feature subset which consists of the features that are more supportive to difficult minority class; (2) construct classification rules based on the original Fish-net algorithm. Our experimental results on Reuters21578 show that the proposed approach achieves better balanced accuracy rate on both majority and minority class than Naive Bayes MultiNomial and SVM.

1 Introduction

Data set imbalance is a commonly encountered problem in text categorization. Given a training set consists of N classes, one of the simplest classification scheme is to build N binary classifier for every individual class. Each classifier will distinguish the instances from one specific topic and all the others. Apparently, in the process of constructing binary classifier, the training set are separated into two sections: the target class, which we will call it minority class; the remaining classes, which we will call it majority class. In this case, whether the classes are evenly distributed in the collection or not, it will easily cause the data set imbalance.

The dimensionality of text data is normally in thousands. Numerous feature selection approaches have been presented in order to eliminate the irrelevant features which can be ignored without degradation in the classifier performance. However, as discussed in [1], most existing methods fail to produce predictive features for difficult class. [1] summarizes the reasons for this as follows:

1. Very few training examples for the class, and/or
2. Lack of good predictive features for that class.

L. Wang and Y. Jin (Eds.): FSKD 2005, LNAI 3614, pp. 39–48, 2005.

The first situation is the instance imbalance. In text classification, along with the instance imbalance, it will also come with the feature imbalance. Assume that we separate the feature set from the majority and minority classes. Since the majority class has a larger number of documents than the minority one, it is more likely to have a larger vocabulary(feature set) than the minority. We call this **Dual Imbalance** and this is an interesting research issue to be looked into.

The research purpose of our work is to improve the classification accuracy on difficult minority class. We present a feature selection method which extracts features supportive to the minority class. Instead of employing traditional classification algorithms, we build the learning scheme based on the field learning strategy.

2 Related Work

Feature selection on imbalanced text data is a relatively new issue in recent literature. In [1], based on the observations, the authors pointed out that existing feature selection mechanisms tend to focus on features that are useful predictors for easier class, while the features for difficult class are easily ignored. Their solution is to apply round-robin turn to let each class propose features. That is, for each class in the data set, rank all features using a certain feature scoring method, such as IG or CHI, and take the best features suggested from each class in turn. Their experiment on some benchmark data set demonstrated consistent improvement for multi-class SVM and Naive Bayes over basic IG or CHI. In [2], given the size of the feature set l, which is pre-defined, positive feature set of size l_1 and negative feature set of size l_2 are generated by ranking the features according to some feature scoring methods. The combination of the positive and negative features is optimized on test or training set by changing the size ratio l_1/l ranging from 0 to 1. Their results show that feature selection could significantly improve the performance of both Naive bayes and regularized logistic regression on imbalanced data.

3 Preliminaries

We use D, to denote a training document set; m, number of total documents; n, number of total terms. We regard each term as a unique attribute for the documents. The definition of head rope is given as follows [3]:

Definition: Head rope
In an $m \times n$ dimension space Ω, a head rope $h_j (1 \leq j \leq n)$ with respect to attribute j consists of the lower and upper bounds of a point set D_j, where $D_j \subseteq \Omega$ is the set of values of the attribute j occur in the instances in the given instance set.

$$h_j = \{h_{l_j}, h_{u_j}\} = \{min_{1 \leq i \leq m}\{a_{ij}\}, max_{1 \leq i \leq m}\{a_{ij}\}\} \tag{1}$$

Let D^+ be the positive document class and D^- be the negative one; h_j is the positive head rope if h_j is derived from D^+. Otherwise, it is the negative

one. Positive and negative head ropes construct the PN head rope pair for an attribute.

The original Fish-Net algorithm [3,4,5] can be summarized as below:

Fish-net Learning Algorithm
Input: A training data set D with a set of class labels $C = \{P, N\}$.
Output: An β-rule which is composed of contribution functions for each attribute, a threshold α and resultant headrope.
1. For each attribute A_j, find out its fields regarding each class.
2. For each attribute A_j, construct its contribution function using its fields.
3. According to the contribution function, work out resultant head rope pair $\langle h^+, h^- \rangle$. For each instance in the training set, we compute the contribution by averaging the contribution values of each attribute. The average contribution of all positive instances compose the positive resultant head rope h^+ and h^- is constructed in the same manner.
4. Determine the threshold α by examining the discovered head rope pair.

The contribution function is used to calculate and measure the contribution of one attribute to the desired class. In [5], the author illustrated six possible relationships between h^+ and h^- as shown in Figure 1.

4 Fish-Net for Text Classification

The original Fish-Net was applied to data set with continuous numeric variables and it is proven to achieve significantly higher prediction accuracy rates than point learning algorithms, such as C4.5. Its training time is linear in both the number of attributes and the number of instances [5]. However, will it still have the high performance on text data? In this section, we will examine the characteristics unbalanced text data has and present our feature selection Fish-net algorithm. Basically, our approach consists of two phases: first, select features supportive to the minorities; second, construct the classification rule based on the original Fish-net.

4.1 Feature Selection on Imbalance Text Data

Table 1 gives a simple example of document-term matrix with two classes. How could we calculate the head rope with 0 values in it? If we take the minimum and maximum value as the lower and upper bound, apparently, a certain number of head ropes will end up beginning with zero. For instance, head rope $[0, 3]$ will be achieved on both classes for *result*. This draws the conclusion that the support of *result* for both classes is similar. Is this the true case? Note that in *cran*, *result* is only contained in one instance while it appears in four instances of *med*. *Result* should have stronger prediction capability for *med* class. Thus, not only we need to consider the value of one attribute, but also we should incorporate its distribution among documents.

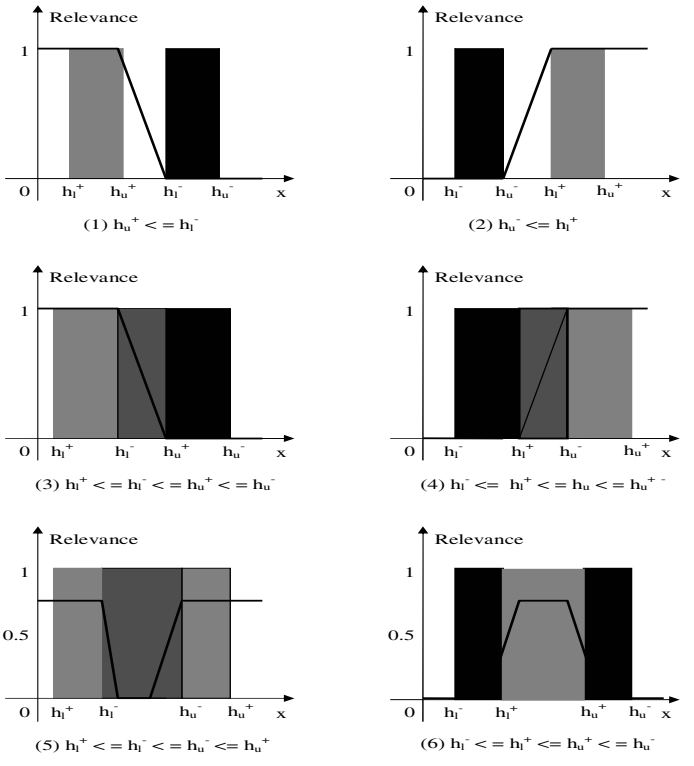

Fig. 1. Six Different Cases for Head Ropes

An alternative way is to calculate the lower bound of the head rope as the average value minus the variance; the upper bound as the average plus the variance. Average indicates the average value of one feature over the entire class and variance indicates how dynamic its distribution in different instances is. However, this approach is not able to detect relevant features in some extreme cases, as shown in the examples below.

Example 1: Suppose that both positive and negative class have 20 instances. Feature A appears in each instance of positive class with frequency 1; in one instance of negative class with frequency 20.

Discussion: *The average value for both positive and negative class is 1. The variance for positive class is 0 while for negative class it is much bigger. Hence, the resulting head rope pair falls in case 6 as in Figure 1. However, this feature is a good predictive feature for positive class from our observation.*

Example 2: The data set is as shown in Table 2.

Discussion: *The average values for both classes are still equal to 1. The resulting head rope pair will either fit in case 5 or 6. Normally features fit in these two cases are regarded as non-informative for both classes and could be discarded.*

Table 1. An Example of Document-feature Data Set

Doc.	flow	form	layer	patient	result	treat
cran.1	1	1	1	0	0	0
cran.2	2	0	1	0	0	0
cran.3	2	1	2	0	3	0
cran.4	2	0	3	0	0	0
cran.5	1	0	2	0	0	0
med.1	0	0	0	8	1	2
med.2	0	1	0	4	3	1
med.3	0	0	0	3	0	2
med.4	0	0	0	6	3	3
med.5	0	1	0	4	0	0
med.6	0	0	0	9	1	1

Table 2. Example 2

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
0	0	5	1	0	0	3	0	0	1

N1	N2	N3	N4	N5
1	2	1	1	0

It appears in four instances of both positive and negative class. However, in the negative class, those four instances comprise the 80% of the entire class while in the positive class, they only comprise 40%. Apparently, this feature could be more supportive to negative class.

In order to overcome these difficulties, we present a varied calculation of the standard average and variance.

$$Average^{'} = \overline{x}^{'} = \frac{Df}{N} \times \frac{Sum}{N} \tag{2}$$

$$Variance^{'} = \varepsilon^{'} = \frac{\frac{Df}{N} \cdot \sum(x_i - \overline{x})^2 + \frac{\overline{Df}}{N} \cdot \sum(x_i - \overline{x})^2}{N - 1} \tag{3}$$

Df is the number of documents contain a feature f in a single class and $\overline{Df}$ is the number of those does not. If f appears in every document, i.e., $Df = N$, then it turns out to be the normal average and variance. Apparently, Df/N reflects the popularity f is in that class and this value is a tradeoff between the feature distribution and its normal average value. If f does not appear in most instances, even its value in the existing ones are high, the average will still be low. The more frequent f is in the class, the higher weight Df/N will give to the normal average.

The variance calculation is based on the following assumption: the instances are separated into those ones with the feature f and those without. The popularity rate Df/N and $\overline{Df}/N$ give weights on the two sections. If f appears in more than half of the instances, then the first part of variance will dominate the final result, otherwise the second part will.

According to the above discussion, the detailed algorithm is described as follows:

Algorithm1: Range-oriented Feature Selection Algorithm
Input: A pre-processed training document matrix with binary class labels {P,N}. The original feature set is F.
Output: A selected feature subset Fs.
1. For each feature $f \in F$, calculate its average and variance in both positive and negative class according to formula (2) and (3).
2. Work out the head rope pair for each feature $f \in F$:

$$h_j^+ = [h_{l_j}^+, h_{u_j}^+] = [\overline{x_j^+} - \varepsilon_j^+, \overline{x_j^+} + \varepsilon_j^+], h_j^- = [h_{l_j}^-, h_{u_j}^-] = [\overline{x_j^-} - \varepsilon_j^-, \overline{x_j^-} + \varepsilon_j^-]$$

3. For each $f \in F$, find out which case its PN head rope pair fits in.
4. Select those features whose PN head rope pair fits in case 2 as in Figure 1. These comprise the feature subset Fs.

4.2 Classification Rule Construction Based on Fish-Net

The second phase of our algorithm is to construct the classification rule on the training data with the selected features. In this section, we will present the detailed algorithm first, then we will further justify our approach. Let I be instance set: $I = I^+ \bigcup I^-$, where I^+ is the positive instance set and I^- is the negative instance set.

Algorithm 2: Improved Fish-Net Algorithm:
Input: The pre-processed training document matrix with selected feature subset Fs.
Output: An β-rule which is composed of contribution functions for each selected attribute, a threshold α and resultant head rope.
1. For each selected feature $f \in Fs$, find out its fields regarding each class as follows:

$$h_j^+ = [h_{l_j}^+, h_{u_j}^+] = [min_{1 \le i \le m}\{a_{ij}(I_i \in I^+)\}, max_{1 \le i \le m}\{a_{ij}(I_i \in I^+)\}(a_{ij} \neq 0)] \tag{4}$$

The same technique applies to derive the negative head rope $h_j^- = [h_{l_j}^-, h_{u_j}^-]$.
2. For each selected feature $f \in Fs$, construct its contribution function using fields $[h_{l_j}^+, h_{u_j}^+]$ and $[h_{l_j}^-, h_{u_j}^-]$.
3. According to the contribution function, work out resultant head rope pair $\langle h^+, h^- \rangle$. For each instance in the training set, we compute the contribution as follows:

$$Contribution = \frac{Sum}{N} * \frac{N}{N_{total}} \tag{5}$$

where Sum is the sum of contribution values of all attributes in each instance; N is the number of non-zero values the instance has in Fs; N_{total} is the number of features(including non-selected ones) the instance has. The positive resultant head rope h^+ is constructed from all positive instances and h^- is constructed from all negative instances.
4. Determine the threshold α by examining the discovered head rope pair.

The first step of the algorithm is to set up the real head rope pair for each selected feature. We calculate the real fields by ignoring all 0 values and taking the minimum and maximum value as the lower and upper bound of the head rope. The reason for us to do this can be seen from the following case study.

Case Study:
Given a data set with 20 positive instances and 500 negative instances.
Feature A appears in all the positive instances and appears in only 20 negative instances. Feature B appears in every positive instance and does not occur in any negative instance. Assume the frequency is 1.

Discussion: *Both Feature A and B will be selected as supportive for minority class. However, if we only consider Feature B in classification, the positive and negative classes can be separated precisely. If only considering Feature A, although most negative instances are classified correctly, there are still 20 negative instances which could possibly be misclassified.*

In other words, among the selected features, there still exists different levels with respect to classification performance. Step1 and 2 in our algorithm helps to further classify the selected features into six cases.

In Step 3, the contribution value for each instance is calculated. In the original approach, it is obtained by averaging the sum of all contribution values. However, this is not feasible in text data. First of all, the number of features a document includes varies and mostly depends on the document length. This easily causes the feature imbalance problem. If we average the sum of contribution values with the total number of features, we will find the longer documents have higher contribution values and this makes shorter documents difficult to classify.

N/N_{total} is the percentage of features selected for classification in an instance. This adds weight to the average contribution value. The reason for this is by considering this situation: in a feature subset, a longer document could possibly have the same amount of features selected as the short ones. However, for the longer document, it could also have a much larger vocabulary which are not selected and more supportive to the majority class. For the short document, the selected features could already be all the words it has.

5 Experimental Work

5.1 Data Set Description

We use **Reuters-21578** Modified Apte ("ModApte") Split to test our algorithm. The collection contains 9603 documents in the training set and 3299 documents in the test set. We preprocessed the documents using the standard stop word removing, stemming and converted the documents to high-dimensional vectors using TFIDF weighting scheme. We choose 10 most frequent topic categories in the experiments. Table 3 summarizes the details. It lists, for each specific topic, the number of positive documents in the training set(#+Training), the number of positive documents in the test set(#+Test). The total number of

Table 3. Reuters-21578 ModApte Dataset Description

Data set	Earn	Acq	Money-fx	Grain	Crude	Trade	Interest	Ship	Wheat	Corn
#+Training	2866	1632	475	371	330	369	347	197	212	181
#+Test	1083	715	151	127	160	117	131	89	71	56

unique terms, the average number of terms per document are staying the same due to the same preprocessing procedure. In order to reduce the size of the term set, we discarded terms which appear in less than 5 documents. The total number of terms extracted finally is 6362 and the average number of terms per document is 41.

5.2 Evaluation Measurement

Table 4 illustrates the contingency table derived from the classification results for a specific category c_i. Note that True Positive Rate(T.P.R. = TP/(TP+FN)) indicates the percentage of correctly classified positive instances in the actual positive class, and False Positive Rate(F.P.R. = FP/(FP+TN)) indicates the percentage of incorrectly classified negative instances in the actual negative class. They are the major measurements we use in our experimental work. We also employ Receiver Operating Characteristic (ROC) curve analysis to characterize the T.P.R. and F.P.R. Accuracy is measured by Area Under Curve(AUC) which refers to the area under the ROC curve. A classifier which can produce the ROC curve with a very sharp rise from $(0, 0)$ and lead to the AUC value close to 1 is regarded as the best.

Table 4. The contingency table for category c_i

C_i	Pos(Standard)	Neg(Standard)
Pos(Classifier)	TP	FP
NegClassifier)	FN	TN

5.3 Experimental Results

The Feature Selection Fish-net learning algorithm is implemented in Java. We compare our approach with Naive Bayes Multinomial implemented in WEKA [6] and SVM in SVMLight [7].

Table 5 reports the classification accuracy on the ten frequent Reuters topics from these three classifiers. The measurement we use is T.P.R. The left column under each classifier is for positive minority class and the right one is for the negative majority class. In general, the classification accuracy of all three learning algorithms on majority class is very high, reaching more than 95% in most cases. But on minority class, the performance varies. For Naive Bayes MultiNomial, the T.P.R. decreases dramatically along with the reduced number of positive instances. On the last five topics, it even has not reached 50%. On each topic's

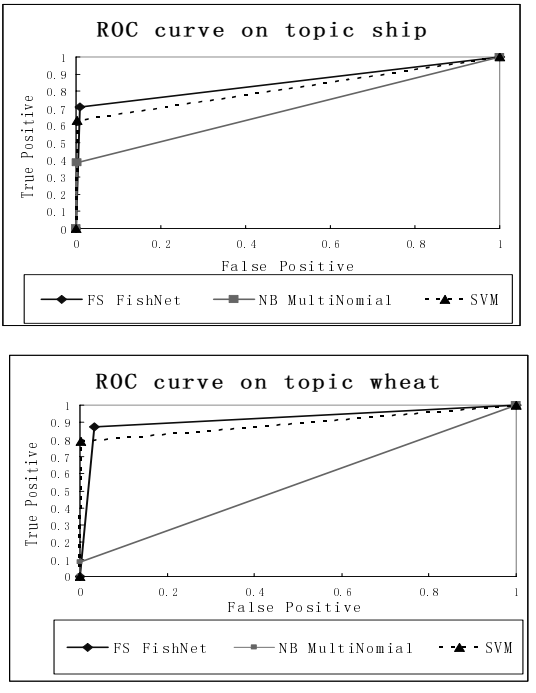

Fig. 2. ROC curves on topic ship and wheat

majority, SVM achieves nearly 100% accuracy rate. However, on minority class, Feature Selection Fish-net achieves better accuracy rate in most cases, especially with small number of positive instances. The accuracy rates of our algorithm on majority and minority are more balanced.

Figure 2 gives the ROC curve obtained on topic *ship* and *wheat* respectively. Apparently, our FS Fish-net performs the best on these three unbalanced text data with larger AUC values.

6 Conclusion

In this paper, we investigate the problem of learning classification rules from dual imbalance text data, which appears to be a common problem in reality. Our approach is designed to improve the classification accuracy on the minority without sacrificing the performance on majority. Our experimental work on the benchmark data set Reuters21578 proves that our approach performs better in achieving balanced accuracy rate than Naive Bayes MultiNomial and SVM.

Our future work will focus on investigating the efficiency issue of the Feature Selection Fish-Net and the possibilities of applying our algorithm to real applications, such as e-mail spam detection, specific target document identification.

Table 5. True Positive Rate on Positive and Negative Class from Feature Selection Fish-Net, Naive Bayes MultiNomial and SVM

Dataset	*FS FishNet*		*NB*		*SVM*	
	P	*N*	*P*	*N*	*P*	*N*
earn	0.874	0.99	0.93	0.992	0.977	0.994
acq	0.866	0.968	0.757	0.997	0.922	0.992
money-fx	0.883	0.956	0.419	0.994	0.698	0.99
grain	0.899	0.947	0.57	0.997	0.879	0.999
crude	0.847	0.935	0.635	0.996	0.836	0.993
trade	0.863	0.898	0.331	1	0.735	0.994
interest	0.756	0.974	0.008	0.999	0.573	0.998
ship	0.708	0.992	0.382	0.998	0.629	0.998
wheat	0.873	0.967	0.085	1	0.789	0.998
corn	0.679	0.972	0.089	1	0.839	0.999

References

1. Forman, G.: A pitfall and solution in multi-class feature selection for text classification. In: Proceedings of the 21st International Conference on Machine Learning. (2004)
2. Zheng, Z., Wu, X., Srihari, R.: Feature selection for text categorization on imbalanced data. ACM SIGKDD Explorations Newsletter :Special issue on learning from imbalanced datasets **6** (2004) 80–89
3. Dai, H., Hang, X., Li, G.: Inexact field learning: An approach to induce high quality rules from low quality data. In: Proceedings of 2001 IEEE International Conference on Data Mining. (2001)
4. Ciesielski, V., Dai, H.: Fisherman: a comprehensive discovery, learning and forecasting systems. In: Proceedings of 2nd Singapore International Conference on Intelligent System. (1994) B297(1)–B297(6)
5. Dai, H., Ciesielski, V.: Learning of inexact rules by the fish-net algorithm from low quality data. In: Proceedings of the Eighth Australian Joint Artificial Intelligence Conference. (1994) 108–115
6. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques with Java implementations. Morgan Kaufmann (1999)
7. Joachims, T.: Making large-scale support vector machine learning practical. In B. Scholkopf, C. Burges, A.S., ed.: Advances in Kernel Methods: Support Vector Machines. MIT Press, Cambridge, MA (1998)