Neighborhood-Hypernetwork for Classification of Imbalanced Data

J. Jiang, H.Q. Ran & K. Yang
College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China

ABSTRACT: There exists several characteristics in imbalanced dataset, such as classes imbalance, between-class imbalance, overlapping, influenced noise, multi-classification with class imbalance, etc., which will greatly influence the classification performance of algorithms on imbalanced datasets. So far, the model has been widely used on many classification problems, such as DNA microarray data, text classification, stock prediction, and so on. As traditional hypernetwork cannot deal with continuous data directly and will bias to majority class when used on imbalanced data, this paper presents a neighborhood-hypernetwork model for classification of imbalanced data. The paper improves the structure of hypernetwork to make sure it can deal with the issues mentioned. The efficiency and advantage of the proposed approaches are verified by simulation experience on the UCI dataset.

KEYWORD: Imbalanced Dataset; Hypernetwork; hypergraph

1 INTRODUCTION

When one class has much more examples than the others we call the problem class imbalance. The imbalanced data problems are widespread in real life. And the study of imbalanced data sets are attracting more and more attention cause imbalanced data sets are becoming ordinary today such as those related to security: spam detection, fraud detection, software defect detection; biomedical data: finding the transition between coding and non-coding DNA in genes, mining cancer gene expression; or financial data, for example, risk predictions in credit data.

The standard classifiers are judged by accuracy, so the minority class examples can be simply ignored. When the data is unbalanced, usually all classifiers present some performance loss. So classification of imbalanced data can be very difficult. Therefore, the traditional classification algorithms do not apply to imbalanced data sets. Ensuring the accuracy of majority examples, developing an improved classification algorithm that can improve accuracy of minority examples has far-reaching significance.

Hypernetwork is a bio-inspired probabilistic graphical model. And it is based on undirected graphs. In this paper, we call a hypergraph whose hyperedges are weighted as a hypernetwork. Unlike the usual graph where an edge can connect two vertexes at most, hyperedges of a hypergraph are able to make connection between two or more vertexes. In the study of hypernetwork, vertexes represent higher-order interactions in the hyperedges. The weight of a hyperedge represents the strength of the association across the vertexes forming the hyperedge. Up to now, hypernetwork models have been successfully used to solve various machine learning problems existed today. Zhang and Jang use a hypernetwork model to diagnose leukemia with micro-array data. Kim et al. perform text classification using a hypernetwork model. Kim and Zhang propose an order variable hypernetwork model for the classifications of the optical recognition of handwritten digits data, the SPECT heart data and the 1984 United States congressional voting records data. Park et al. use a hypernetwork model to mine potential prostate related macro-biomarkers. Kim et al. use a hypernetwork for micro-RNA expression profile analysis. Kim et al. proposed a mutual information based hypernetwork model for brain data analysis to find potential significant modules on IQ from brain MRI data.

2 HYPERNETWORK MODEL

2.1 *The basic model of a hypernetwork*

Hypernetwork is a probabilistic graphical model. A hypernetwork represents higher-order relationships between different factors. A hypernetwork is described as a triple H= (X, L, W), where X=(x1, x2…xn) is the set of feature variables. And L= {l1, l2, l3… lm} represents the hyperedges library. A hyperedge contains feature variables and class label it belongs to. This paper uses W to describe the weight of a hyperedge. The paremeter corresponds to the number of copies of a hyperedges. W also represents the degree of correlation among its vertices. Vertices of a hyperedge represent the cardinal number or sequence of this hyperedge. Here is an example, li= (xi1, xi2…xik, yi), 1≤k≤n. This is a k order hyperedge. In this case, n represents the whole number of features of this dataset and yi represents its class label. Fig.1 desceibes an example of a hypernetwork. In Fig.1, every oval is a hyperedge. And the thicker the line is, the more strength it weights.

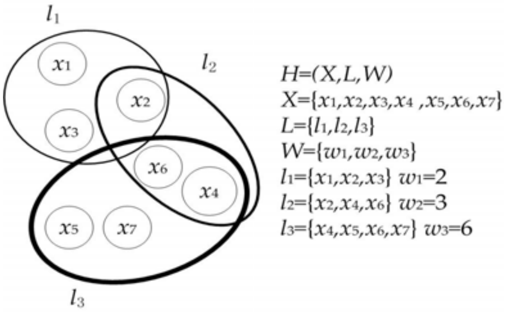

l_1
l_2
l_3
x_1
x_2
x_3
x_6
x_4
x_5
x_7
$H=(X,L,W)$
$X=\{x_1,x_2,x_3,x_4 ,x_5,x_6,x_7\}$
$L=\{l_1,l_2,l_3\}$
$W=\{w_1,w_2,w_3\}$
$l_1=\{x_1,x_2,x_3\}$ $w_1=2$
$l_2=\{x_2,x_4,x_6\}$ $w_2=3$
$l_3=\{x_4,x_5,x_6,x_7\}$ $w_3=6$

Fig.1. A hypernetwork with 7 vertices and 3 hyperedges

Hypernetworks are strongly related to the probabilistic library models (PLM) (Zhang and Jang, 2005b) that are simulated by a DNA computing model (Ha et al, 2007). The PLMs are considered as the prototype of hypernetworks and hypernetworks are considered as the phenotype of PLMs. In this paper, hypernetwork is considered as a decision rule based classifier wherein a hypernetwork corresponds to a collection of decision rules, a hyperedge corresponds to a decision rule, and a vertex of a hyperedge corresponds to an attribute in the training data. And a weight reflects numbers of copies of a hyperedge (decision rule).

2.2 *Hypernetwork as a classifier*

Running as a classifier, H is used to describe the relation between the system inputs and outputs in a hypernetwork, i.e. H: X→Y ($X \subseteq R^d$ and $Y \subseteq R$). In this paper, sampling n times the set X ×Y we can get the training data set $D=\{(\mathbf{x}_i, y_i) \in X \times Y\}_{i=1}^{n}$. The set is according to an independent and identical distribution P(X,y). The probability distribution P(X, y) shows the corresponding relationships between inputs and outputs. If we get a P(X, y), we can get the relationship between inputs X to output y as (1). However, the probability distribution P(X, y) is unknown, and only some samples, the training set D, are available.

$$P(y \mid X) = \frac{P(X, y)}{P(X)} \tag{1}$$

When using a hypernetwork as classifier, the hyperedges can show the probability distribution P(X, y) .Firstly, the training examples are stored and each of them has multi-copies. The number of copies of library elements is updated as new training examples are observed so that their frequency is proportional to their probability of observation. In essence, libraries of hyperedge represent the joint probability between input and output.

$$P(X, y) \approx \frac{1}{|L|} \sum_{i=1}^{|L|} f_i^n(x_1, x_2, ..., x_n, y) \tag{2}$$

In (2), $f_i^n(x_1, x_2, ..., x_n, y_i)$is the hyperedge. n and |L| represent the size of hyperedge library.

Fig.2 shows a hypernetwork classifier process. First, the feature data is mapped to the hyperedge vertices trainning attribute. And the class attributes is used as the training data labels of hyperedges. These hyperedges can be set as some decision rules. We can use these rules mapping the observation of vertices' values to appropriate action (class label). So the hyperedges library can represent the relationship between the input X and the output label y∈{-1, +1}as P (X, y).

Fig.3 shows how the hypernetwork works during the process of classification. Given an input sample X, all hyperedges matching with X are extracted from the library L. The extraction is implemented by matching X with each and every library element of the library L. Then the classifier calculates the probability of each class y, the class decision y* is made by selecting the class which has the highest probability.

$$y^* = \arg \max_{y \in \{-1,+1\}} \left(\frac{P(X, y)}{P(X)} \right) = \arg \max_{y \in \{-1,+1\}} \left(P(y \mid X) \right) \tag{3}$$

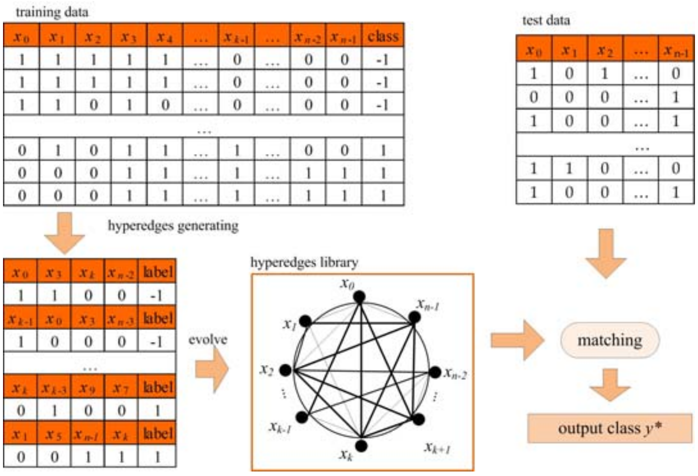
In step 3.1, X is the count in M. And it is used to denote approximates the probability of observing the matching individuals:

$$count(X) / |L| = |M| / |L| \approx P(X) \tag{4}$$

Step 3.2 and 3.3 figures the rate of recurrence denoted by count (y|X) of each class, and y∈{-1,+1}is the class label. Count (y| X) shows an approximation of a posteriori probability. And this value is the conditional probabilities of the example:

$$count(y \mid X) / |L| = |M^y| / |M| \approx P(y \mid X) \tag{5}$$

Step 4 makes decision by comparing the posteriori probability of each class to which X belongs. The class decision y* is made by selecting the class which has the highest probability.



training data

x_0	x_1	x_2	x_3	x_4	...	x_{k-1}	...	x_{n-2}	x_{n-1}	class
1	1	1	1	1	...	0	...	0	0	-1
1	1	1	1	1	...	0	...	0	0	-1
1	1	0	1	0	...	0	...	0	0	-1
...										
0	1	0	1	1	...	1	...	0	0	1
0	0	0	1	1	...	1	...	1	1	1
0	0	0	1	1	...	1	...	1	1	1

test data

x_0	x_1	x_2	...	x_{n-1}
1	0	1	...	0
0	0	0	...	1
1	0	0	...	1
...				
1	1	0	...	0
1	0	0	...	1

x_0	x_3	x_k	x_{n-2}	label
1	1	0	0	-1
x_{k-1}	x_0	x_3	x_{n-3}	label
1	0	0	0	-1
...				
x_k	x_{k-3}	x_9	x_7	label
0	1	0	0	1
x_1	x_5	x_{n-1}	x_k	label
0	0	1	1	1

Fig.2. Flow chart of the evolution process of a hypernetwork classifier

```
1. Let the library $L$ represent the current empirical distribution $P(X,y)$
2. Given an input $X$
3. Classify $X$ using $L$ as follows:
    3.1 Extract all hyperedges matching with $X$ into $M$
    3.2 Count the number of hyperedges whose label is -1 into $M^1$
    3.3 Count the number of hyperedges whose label is 1 into $M^1$
4. Make descion $y^*$. If $|M^1|>|M^1|$,$y^*$=-1 else $y^*$=1;
```

Fig.3. Classification course of a hypernetwork

3 NEIGHBORHOOD-HYPERNETWORK

Generally the neighborhood is defined as maximum distance between centers to border in a metric. Given an N-dimensional real numbers space $\mathbf{R}$, $d:\mathbf{R}^N \times \mathbf{R}^N \to \mathbf{R}$,d is called a metric on the space $\mathbf{R}$ .if d satisfies the following conditions:

(1) $d(\mathbf{x}_1,\mathbf{x}_2) \geq 0, d(\mathbf{x}_1,\mathbf{x}_2)=0$ when $\mathbf{x}_1 = \mathbf{x}_2$, $\forall \mathbf{x}_1,\mathbf{x}_2 \in \mathbf{R}^N$;

(2) $d(\mathbf{x}_1,\mathbf{x}_2) = d(\mathbf{x}_2,\mathbf{x}_1), \forall \mathbf{x}_1,\mathbf{x}_2 \in \mathbf{R}^N$;

(3) $d(\mathbf{x}_1,\mathbf{x}_3) \leq d(\mathbf{x}_1,\mathbf{x}_2) + d(\mathbf{x}_2,\mathbf{x}_3), \forall \mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3 \in \mathbf{R}^N$.

The <R, d> is called a metric space. The Minkowski distance is defined as measure function in the formula (6):

$$d_{\mathbf{P}}^{\lambda}(\mathbf{x}_i,\mathbf{x}_j) = \left[\sum_{k=1}^{|\mathbf{P}|} \left| f(\mathbf{x}_i,a_k) - f(\mathbf{x}_j,a_k) \right|^{\lambda} \right]^{1/\lambda} \tag{6}$$

The neighborhoods of all the objects divide domain into a plurality of basic information particle. Fig.4 shows a sample ε-space neighborhood.

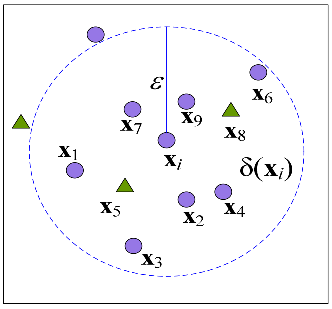


Fig.4. ε-neighborhood of $\mathbf{x}_i$

Neighborhood-hypernetwork works as a decision-making system. Hyperedges show sample distribution to the class field mapping of the category. Samples belonging to the same areas are highly similar. The adjacent samples are probably belonging to the same class. According to the theory of statistics, measuring the neighborhood of sample class distribution, it is classified as the bigger weight class. The basic idea is described as follows:

①Data preprocessing

So as to eliminate the impact dimension, the data normalization is made. Using the conventional maximum and minimum normalization method, the formula is as following:

$$x_{ij}^{'} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{7}$$

$x_{ij}^{'}$ is the transformed value of x_{ij}, $\min(x_j)$ and $\max(x_j)$ represent the minimum and maximum property of x_j.

②Initialization neighborhood-hypernetwork.

Given training set D, for each sample xi in training set, the neighborhood radius ε ranges from $[r_{\min}, r_{\max}]$. Neighborhood radius is shown in Fig.5, which is calculated according to formula (8) and (9):

$$r_{\min} = u \cdot \min(d_{\mathbf{B}}(\mathbf{x},\mathbf{x}_i)) \tag{8}$$

$$r_{\max} = \min(d_{\mathbf{B}}(\mathbf{x},\mathbf{x}_i)) + v \cdot [\max(d_{\mathbf{B}}(\mathbf{x},\mathbf{x}_i)) - \min(d_{\mathbf{B}}(\mathbf{x},\mathbf{x}_i))] \tag{9}$$

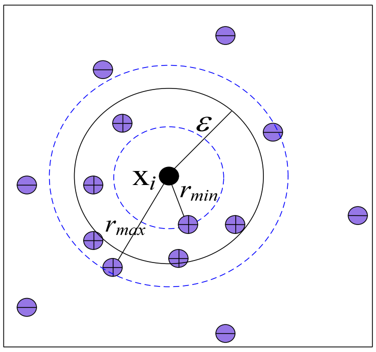


Fig.5. Radius of neighborhood hypernetwork

4 EXPERIMENT AND ANALYSIS

4.1 *Experimental datasets and parameters setting*

The experimental data sets are 9 commonly used imbalanced data sets collected from the UCI. Table 4.1 describes the relevant information of the data sets. This paper select one of the categories as minority class and others as majority class. Parameters are shown in Table 4.2.

Table 4.1 Data description

Data set	Min vs.Maj	Attributes	Samples	Categories	Categories distribution
Glass1	1,2 vs. others	9C	214	6	68/146
Glass7	7 vs. others	9C	214	6	29/185
Ecoli4	Imu vs. others	7C	336	8	35/301
Iris	Iris-virginica vs. others	4C	150	3	50/100
Seed	2 vs. others	7C	210	3	70/140
Wine	3 vs. others	13C	178	3	48/130
Yeast	CYT vs. POX	8C	483	10	20/463
Haberman	2 vs. 1	3C	306	2	81/225
Transfusion	1 vs. 0	4C	748	2	178/500

4.2 *The classification results comparison*

Table 4.3 Value of G-mean

Data set	G-mean			
Glass1	Smote+KNN	SMOTE-RSB+C4.5	N-HN	W-N-HN
	0.62	0.7938	0.7337	**0.8215**
Glass7	SVM	SMOTE-Boost	N-HN	W-N-HN
	0.8660	0.911	0.8394	**0.9265**
Ecoli4	Ada.M1	SMOTE+Ada.M1	N-HN	W-N-HN
	0.790	0.795	0.5989	**0.8974**
Iris	Smote+KNN	SMOTE-RSB+C4.5	N-HN	W-N-HN
	0.9105	0.9349	**0.9409**	0.9278
Seed	SVM	SMOTE-RSB+C4.5	N-HN	W-N-HN
	0.6473	**0.9749**	0.9423	0.9672
Wine	Smote+J4.8	DB_Smote+J4.8	N-HN	W-N-HN
	0.8805	0.890	0.8256	**0.9558**
Yeast	Adacost	GSVM-RU	N-HN	W-N-HN
	0.809	0.798	0.90	**0.9230**
Haberman	RUSBoost+C4.5	EasyEnsemble+C4.5	N-HN	W-N-HN
	0.5889	0.623	**0.6363**	0.564
Transfusion	RUSBoost+C4.5	EasyEnsemble+C4.5	N-HN	W-N-HN
	0.6254	0.674	0.631	**0.7325**

Table 4.4 Value of F-measure

Data set	F-measure			
Glass	RSBoundary+C4.5	SMOTE-RSB+C4.5	N-HN	W-N-HN
	0.7214	0.7419	0.6851	**0.7980**
Glass7	DataBoost-IM	SMOTEBoost	N-HN	W-N-HN
	0.892	0.84	0.7867	0.7971
Ecoli4			N-HN	W-N-HN
			0.4786	**0.5909**
Iris	Smote+KNN	SMOTE-RSB+C4.5	N-HN	W-N-HN
	0.9203	**0.9349**	0.9255	0.9030
Seed	SMOTE-RSB+C4.5		N-HN	W-N-HN
	0.9645		0.9286	0.9538
Wine	Smote+J4.8	DB_Smote+J4.8	N-HN	W-N-HN
	0.80	0.8012	0.7940	**0.8905**
Yeast	DB-IM	GSVM-RU	N-HN	W-N-HN
	0.580	0.688	0.8167	**0.8637**
Haberman	RUSBoost+C4.5	EasyEnsemble+C4.5	N-HN	W-N-HN
	0.4487	0.468	0.4882	**0.5161**
Transfusion	RUSBoost+C4.5	EasyEnsemble+C4.5	N-HN	W-N-HN
	0.4832	0.501	0.4596	**0.5531**

As shown, the neighborhood hypernetwork has obvious advantages on 3metrics. The values of G-means have obvious improvement on Glass1, Glass 7, Ecoli and Seed datasets.

5 CONCLUSION

As traditional hypernetwork cannot deal with continuous data directly and will bias to majority class when used on imbalanced data, this paper presents a neighborhood-hypernetwork model for classification of imbalanced data. The paper improves the structure of hypernetwork to make sure it can deal with the issues mentioned. The efficiency and advantage of the proposed approaches are verified by simulation experience on the UCI dataset.

ACKNOWLEDGEMENTS

This work was supported by the Postgraduate Research and Innovation Project of Chongqing under Grant CYS15167, CYS14150, CYS14147.

REFERENCES

G.-G.Geng, C.-H.Wang, Q.-D.Li, L.Xu, X.-B.Jin. 2007. Boosting the performance of web spam detection with ensemble under-sampling classification, Fuzzy Systems and Knowledge Discovery, Vol4:583-587.

H.Yu, J.Ni, Y.Dan, S.Xu, 2012. Mining and integrating reliable decision rules for imbalanced cancer gene expression data sets, Tsinghua Sci.Tech, 17(6): 666-673.

M.Di Martino, F. Decia, J. Molinelli, A.Fernandez. 2012. Improving electric fraud detection using class imbalance strategies, ICPRAM, vplume 2:135-141.

N.Garcia-Pedrajas, J.perez-Rodriguez, M.D.Garcia-Pedrajas, D.Ortiz-Boyer, C.Fyfe. 2012. Class imbalance methods for translation initiation site recognition in DNA sequences, Knowledge-Based Sys, 25(1):22-34. R.C.Prati, G.E.Batista, D.F.Silva. 2014. Class imbalance revisited: a new experimental setup toassess the performance of treatment methods, Knowledge and Information Systems, pp: 1-24.

S.Wang, X.Yao. 2013. Using class imbalance learning for software defect prediction, IEEE Transactions on Reliability, 62(2):434-443.

V.Garcia, A.I. Marques, J.S. Sanchez. 2012. Improving risk predictions by preprocessing imbalanced credit data, Neural Information Processing, Springer: 68-75.

V.Sofia.2005.Issues in mining imbalanced data sets, Proceedings of the Sinteen Midwest Artificial Intelligence and Cognitive Science Conference, pp: 67-73.