PHÂN LỚP DỮ LIỆU KHÔNG CÂN BẰNG VỚI ROUGHLY BALANCED BAGGING

Phan Bích Chung[1] và Đỗ Thanh Nghị[2]

ABSTRACT

In this paper, we present a novel improvement of the Roughly Balanced Bagging algorithm (Hido & Kashima, 2008) to deal with the imbalanced data classification. Our proposal use ensemble-based algorithms including Boosting (Freund & Schapire, 1995), Random forest (Breiman, 2001) as base leaner of the orginal Roughly Balanced Bagging instead of a single decision tree (Quinlan, 1993). In addition, the distribution in each subset determined by under-sampling of the majority class is belongs to negative binomial distribution function using adjust parameter. The experimental results on imbalanced datasets from UCI repository (Asuncion & Newman, 2007) showed that our proposal outperforms the orginal Roughly Balanced Bagging.

Keywords: Imbalanced data, Roughly Balanced Bagging, Bagging, Boosting, AdaBoost, Random Forest, Decision Tree, Negative binomial distribution

Title: Classification of imbalanced data with roughly balanced bagging

TÓM TẮT

Trong bài báo này, chúng tôi trình bày một cải tiến của giải thuật Roughly Balanced Bagging (Hido & Kashima, 2008) cho việc phân lớp các tập dữ liệu không cân bằng. Chúng tôi đề xuất sử dụng các giải thuật tập hợp mô hình bao gồm Boosting (Freund & Schapire, 1995), Random forest (Breiman, 2001), làm mô hình học cơ sở của giải thuật Roughly Balanced Bagging gốc, thay vì sử dụng một cây quyết định (Quinlan, 1993). Chúng tôi cũng đề xuất điều chỉnh cách lấy mẫu giảm phần tử lớp đa số theo hàm phân phối nhị thức âm ở mỗi lần. Kết quả thực nghiệm trên các tập dữ liệu không cân bằng được lấy từ nguồn UCI (Asuncion & Newman, 2007) cho thấy rằng phương pháp mà chúng tôi đề xuất cho hiệu quả phân loại chính xác hơn khi so sánh với giải Roughly Balanced Bagging gốc.

Từ khoá: Dữ liệu không cân bằng, Roughly Balanced Bagging, Bagging, Boosting, AdaBoost, Rừng ngẫu nhiên, Cây quyết định, Phân phối nhị thức âm

1 GIỚI THIỆU

Phân lớp dữ liệu không cân bằng là một trong 10 vấn đề khó đang được cộng đồng máy học và khai mỏ dữ liệu quan tâm (Yang & Wu, 2006). Vấn đề không cân bằng lớp thường xảy ra với bài toán phân lớp nhị phân (chỉ có 2 lớp) mà ở đó một lớp mà người ta quan tâm chiếm tỉ lệ rất nhỏ so với lớp còn lại. Trong nhiều ứng dụng thực tế, chẳng hạn như phát hiện các giao dịch gian lận, phát hiện xâm nhập mạng, sự rũi ro trong quản lý, phân loại văn bản hay chẩn đoán trong y học. Sự không cân bằng lớp nó ảnh hưởng rất lớn đến hiệu quả của các mô hình phân loại. Ví dụ, trong lĩnh vực phát hiện sự xâm nhập mạng, số lượng các xâm nhập trên mạng thường là một phần rất nhỏ trong tổng số các giao dịch mạng. Hay trong cơ sở dữ

[1] Trường THPT Lê Lợi, Số 19 – Đường Tôn Đức Thắng, Phường 6 – Tp. Sóc Trăng
[2] Khoa CNTT&TT, Trường Đại học Cần Thơ

liệu y học, khi phân loại các pixels trong các ảnh phim chụp tia X có bị ung thư hay không, những pixels không bình thường (ung thư) chỉ chiếm một phần rất nhỏ trong toàn bộ ảnh. Với các tập dữ liệu của các bài toán phân lớp như vậy sẽ làm cho các mô hình học phân lớp gặp rất nhiều khó khăn trong dự báo cho dữ liệu lớp thiểu số. Hầu hết giải thuật học như cây quyết định C4.5 (Quinlan, 1993), CART (Breiman *et al*., 1984), SVM (Vapnik, 1995) đều được thiết kế để cho độ chính xác tổng thể, không quan tâm đến bất kỳ lớp nào. Chính vì lý do này, các giải thuật học phân lớp cho tập dữ liệu không cân bằng gặp phải vấn đề dự báo dễ làm mất lớp thiểu số mặc dù cho độ chính xác tổng thể rất cao. Ví dụ như tập dữ liệu cho dự báo bệnh **A** có 40000 phần tử, trong đó lớp bệnh **A** là lớp thiểu số (người ta quan tâm hay lớp dương) chỉ có 10 phần tử và lớp còn lại (không bệnh, lớp âm) có 39990 phần tử. Một giải thuật học dự báo sai hoàn toàn bệnh **A** (*lúc nào cũng dự báo là không bị bệnh **A***) thì vẫn cho độ chính xác tổng thể là *99,975%*. Đây là một trong những sai lầm nghiêm trọng của giải thuật học phân lớp. Chính vì lý do này, cộng đồng máy học cũng đã tập trung để giải quyết vấn đề phân lớp dữ liệu không cân bằng, chẳng hạn như các cuộc hội thảo khoa học (Chawla *et al*., 2003, 2004).

Nhiều giải pháp cũng đã được đề xuất để giải quyết vấn đề trên trong giải thuật học cây quyết định nhằm cải thiện dự báo lớp thiểu số nhưng không làm mất nhiều dự báo lớp đa số. Chiến lược thay đổi phân bố dữ liệu, (Chawla *et al*., 2003) đề xuất phương pháp lấy mẫu tăng thêm cho lớp thiểu số. (Liu *et al*., 2006), (Hido & Kashima, 2008) đề xuất lấy mẫu giảm cho lớp đa số. Chiến lược can thiệp trực tiếp giải thuật học cây quyết định, (Lenca *et al*., 2008) đề xuất thay đổi hàm phân hoạch dữ liệu nhằm cải thiện dự báo lớp thiểu số nhưng không làm mất nhiều dự báo lớp đa số. (Domingos, 1999), (Weiss & Provost, 2003) đề xuất gán giá phải trả cho dự báo sai của các lớp khác nhau (giá của lớp thiểu số lớn hơn giá của lớp đa số). Ngoài ra (Domingos, 1999) đề xuất điều chỉnh ước lượng xác suất tại nút lá của cây nhằm cải tiến dự báo lớp thiểu số.

Chúng tôi đề xuất cải tiến giải thuật Roughly Balanced Bagging – RB Bagging (Hido & Kashima, 2008) giúp cải thiện dự báo lớp thiểu số nhưng không làm mất quá nhiều dự báo lớp đa số. Chúng tôi đề xuất thay thế mô hình học cơ sở là cây quyết định C4.5 (Quinlan, 1993) dùng trong RB Bagging bằng giải thuật tập hợp mô hình mạnh hơn như rừng ngẫu nhiên (Breiman, 2001) và AdaBoost (Freund & Schapire, 1995). Ngoài ra, chúng tôi cũng điều chỉnh cách lấy mẫu giảm phần tử lớp đa số theo hàm phân phối nhị thức âm ở mỗi lần của RB Bagging để không làm mất quá nhiều dự báo lớp đa số. Kết quả thực nghiệm trên 10 tập dữ liệu không cân bằng từ kho dữ liệu UCI (Asuncion & Newman, 2007) cho thấy rằng phương pháp mà chúng tôi đề xuất (RB Bagging cải tiến) cho hiệu quả cao hơn khi so sánh với giải RB Bagging gốc, dựa trên các tiêu chí về precision, recall, F1-measure và accuracy (van Rijsbergen, 1979).

Phần tiếp theo của bài báo được tổ chức như sau: Trong phần 2, chúng tôi sẽ trình bày ý tưởng chính của giải thuật RB Bagging và giải thuật cải tiến cho phân lớp dữ liệu không cân bằng. Tiếp theo là các kết quả thực nghiệm được trình bày trong phần 3 trước khi kết luận và hướng phát triển được trình bày ở phần 4.

2 GIẢI THUẬT RB BAGGING CẢI TIẾN

Giải thuật RB Bagging của (Hido & Kashima, 2008) cho vấn đề phân lớp dữ liệu không cân bằng với chiến lược thay đổi phân bố của tập dữ liệu. RB Bagging tập trung cải thiện hiệu quả dự báo cho lớp thiểu số, bên cạnh đó cố gắng đảm bảo sử dụng hầu hết thông tin cho lớp đa số. Để đạt được mục tiêu này, RB Bagging cố gắng cân bằng phân bố dữ liệu của lớp dương (lớp thiểu số mà người ta quan tâm) và lớp âm (lớp còn lại). Giải thuật RB Bagging như trình bày trong giải thuật 1 có thể được trình bày ngắn gọn như sau.

Giả sử tập dữ liệu không cân bằng $\boldsymbol{D}$ bao gồm $\boldsymbol{N^{pos}}$ phần tử lớp dương trong $\boldsymbol{D^{pos}}$ và $\boldsymbol{N^{neg}}$ phần tử lớp âm trong $\boldsymbol{D^{neg}}$. RB Bagging thực hiện xây dựng tập hợp $\boldsymbol{K}$ cây quyết định C4.5 (Quinlan, 1993). Ở mỗi bước lặp $\boldsymbol{k}$, RB Bagging thực hiện chiến lược lấy mẫu giảm trên lớp đa số (lớp âm) và toàn bộ lớp thiểu số (lớp dương) để xây dựng mô hình cơ sở cây quyết định. RB Bagging sử dụng số lượng phần tử lớp dương (lớp thiểu số) bằng với số phần tử lớp dương $\boldsymbol{N^{pos}}$ trong tập dữ liệu $\boldsymbol{D^{pos}}$. Nếu chúng được lấy mẫu không hoàn lại, thì tất cả các phần tử lớp dương $\boldsymbol{D^{pos}}$ sẽ được sử dụng trong tập học. RB Bagging sử dụng lấy mẫu giảm số phần tử lớp đa số (lớp âm) với số lượng được xác định theo phân phối nhị thức âm trong đó các tham số là số lượng phần tử bằng với $\boldsymbol{N^{pos}}$ của lớp thiểu số (dương) và xác suất thành công $\boldsymbol{q=0,5}$. Điểm chính là số lượng phần tử của cả hai lớp được lấy với xác suất bằng nhau, nhưng chỉ có kích thước của lớp đa số (lớp âm) thay đổi và số lượng phần tử lớp dương được giữ nguyên vì chúng rất nhỏ. Tập mẫu vừa tạo được dùng để xây dựng mô hình cây quyết định ở bước lặp $\boldsymbol{k}$. Trong dự báo một phần tử mới đến dựa trên chiến lược bình chọn số đông từ kết quả dự báo của $\boldsymbol{K}$ mô hình cơ sở cây quyết định.

Trong xử lý vấn đề không cân bằng lớp, chiến lược lấy mẫu của RB Bagging có thể hiểu như một việc lấy mẫu lặp lại từng mẫu một mà việc chọn lớp được lấy mẫu dựa trên xác suất tiên nghiệm cân bằng $\boldsymbol{p(dương) = p(âm) = 0,5}$. Trên thực tế, phương pháp này tương đồng với việc lấy mẫu bootstrap (lấy mẫu ngẫu nhiên có hoàn lại) mà ở đó kích thước mẫu của mỗi lớp được chọn theo phân phối nhị thức âm với $\boldsymbol{p(dương) = p(âm) = 0,5}$. Mặc dù kích thước của những tập con hơi khác so với nhau, nhưng hầu hết được cân bằng trên bình quân.

Kết quả thực nghiệm trong (Hido & Kashima, 2008) cho thấy rằng RB Bagging sử dụng chiến lược lấy mẫu dựa trên phân phối nhị thức âm đảm bảo chất lượng của giải thuật Bagging gốc của (Breiman, 1996) nhưng sử dụng được hầu hết thông tin của lớp thiểu số. Tuy nhiên, vì quá quan tâm đến lớp thiểu số nên có thể không đảm bảo được kích thước mẫu của lớp đa số trong mỗi lần lặp, nó sẽ làm mất đi một số lượng lớn thông tin (có thể quan trọng) trong lớp đa số. Điều này dẫn đến việc RB Bagging cải tiến được dự báo lớp thiểu số (dương) nhưng lại làm mất dự báo lớp đa số (âm). Để khắc phục khuyết điểm của RB Bagging gốc, chúng tôi đề xuất giải thuật cải tiến vẫn giữ được hiệu quả dự báo lớp thiểu số như RB Bagging gốc nhưng không làm mất nhiều dực báo lớp đa số.

- Đầu vào:

D tập dữ liệu không cân bằng

L **giải thuật cơ sở (cây quyết định C4.5)**

K số bước lặp

- Xây dựng mô hình phân lớp RB Bagging:

Chia tập dữ liệu D thành tập dữ liệu lớp âm D^{neg} và lớp dương D^{pos}

Cho $k = 1$ tới K

Xác định số lượng phần tử lớp âm N_k^{neg} từ phân phối nhị thức âm với tham số như $\boldsymbol{n = N^{pos}}$ và xác suất thành công $q = 0.5$

Lấy số phần tử lớp dương N_k^{pos} asbằng với kích thước lớp dương từ D^{pos}

Thực hiện lấy mẫu ngẫu nhiên N_k^{neg} từ tập D^{neg} tạo ra D_k^{neg}

Thực hiện lấy mẫu ngẫu nhiên N_k^{pos} từ tập D^{pos} tạo ra D_k^{pos}

Xây dựng mô hình $f^k(x)$ bằng giải thuật cơ sở L trên tập dữ liệu bao gồm D_k^{neg} và D_k^{pos}

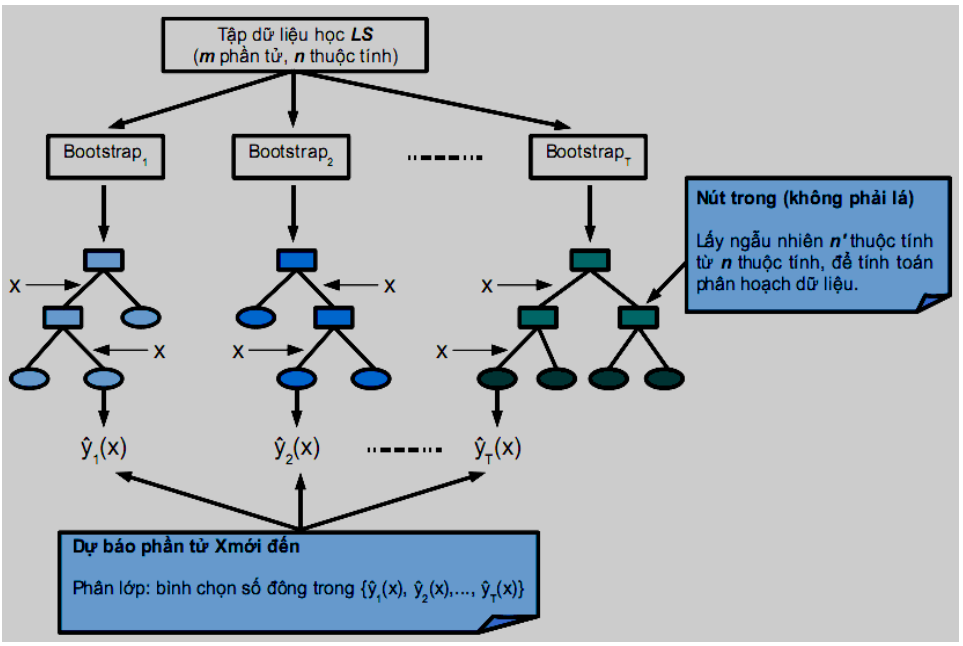
- Dự báo phần tử x_i mới đến:

Bình chọn số đông của các $\{f^k(x_i)\}_{k=1,K}$

Giải thuật 1: Giải thuật RB Bagging

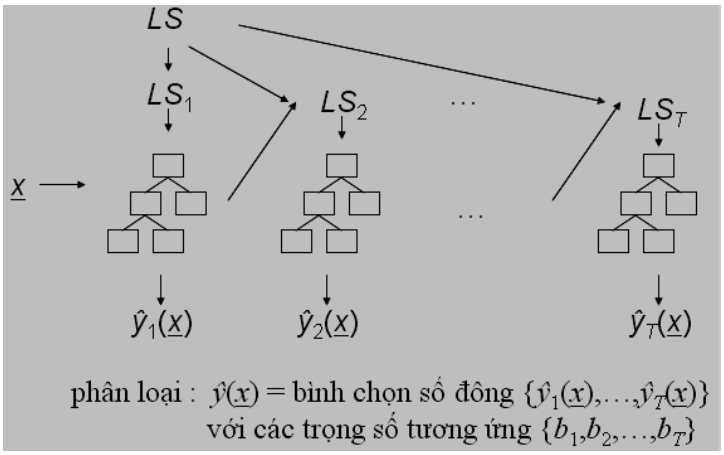
Từ giải thuật RB Bagging gốc, chúng tôi điều chỉnh lại cách lấy mẫu giảm của lớp âm và thay thế mô hình học cơ sở cây quyết định bằng phương pháp tập hợp mô hình. Do RB Bagging gốc lấy mẫu giảm quá nhiều lớp đa số ở mỗi lần lặp (chỉ sử dụng số lượng tương đương với lớp thiểu số) gây ra dự báo lệch quá nhiều sang lớp thiểu số và giảm đáng kể dự báo lớp đa số. Trong cải tiến, chúng tôi đề nghị sử dụng lấy mẫu giảm lớp đa số cũng dựa trên phân phối nhị thức âm nhưng với tham số $n = \sqrt{\frac{N^{neg}}{N^{pos}}} N^{pos}$ thay vì là $n = N^{pos}$ như trong RB Bagging gốc. Ngoài ra, để nâng cao hiệu quả của dự báo, chúng tôi cũng đề xuất sử dụng phương pháp tập hợp mô hình như rừng ngẫu nhiên (Breiman, 2001) và AdaBoost (Freund & Schapire, 1995) làm mô hình học cơ sở mạnh hơn mô hình đơn cây quyết định ở mỗi bước lặp của RB Bagging.

Rừng ngẫu nhiên (giải thuật 2) tạo ra một tập hợp các cây quyết định không cắt nhánh, mỗi cây được xây dựng trên tập mẫu bootstrap (lấy mẫu ngẫu nhiên có hoàn lại), tại mỗi nút phân hoạch tốt nhất được thực hiện từ việc chọn ngẫu nhiên một tập con các thuộc tính. Lỗi tổng quát của rừng phụ thuộc vào độ chính xác của từng cây thành viên trong rừng và sự phụ thuộc lẫn nhau giữa các cây thành viên. Giải thuật rừng ngẫu nhiên cho độ chính xác cao khi so sánh với các thuật toán học có giám sát hiện nay, chịu đựng nhiễu tốt.



Giải thuật 2: Giải thuật rừng ngẫu nhiên

Giải thuật AdaBoost (giải thuật 3) xây dựng tuần tự ***T*** mô hình, lặp lại quá trình học của một mô hình phân lớp yếu (cây quyết định) ***T*** lần. Sau mỗi bước lặp, mô hình phân lớp yếu (cây quyết định) sẽ tập trung học trên các phần tử bị phân lớp sai bởi các lần trước. Để làm được điều này, cần gán cho mỗi phần một trọng số. Khởi tạo, trọng số của các phần tử bằng nhau. Sau mỗi bước học, các trọng số này sẽ được cập nhật lại (tăng trọng số cho các phần tử bị phân lớp sai và giảm trọng số với các phần tử phân lớp đúng). Đặt trọng số cho các mô hình dựa trên lỗi của các mô hình cơ sở. Kết thúc giải thuật sẽ dùng chiến lược bình chọn số đông với trọng số để phân lớp phần tử dữ liệu.



Giải thuật 3: Giải thuật AdaBoost

Việc điều chỉnh cách lấy mẫu giảm của lớp đa số và thay thế mô hình cơ sở bằng phương pháp tập hợp mô hình vì thế giúp cho giải thuật RB Bagging cải tiến mà chúng tôi đề xuất, xử lý tốt hơn khi phân lớp tập dữ liệu không cân bằng vì giữ được hiệu quả dự báo lớp thiểu số như giải thuật RB Bagging gốc nhưng vẫn đảm bảo không làm mất nhiều thông tin của lớp đa số. Tuy nhiên, thời gian thực thi của nó lâu hơn so với giải thuật RB Bagging gốc.

3 KẾT QUẢ THỰC NGHIỆM

Để đánh giá hiệu quả của giải thuật RB Bagging cải tiến, chúng tôi tiến hành cài đặt tất cả chương trình bằng ngôn ngữ **R** (Ihaka & Gentleman, 1996). Thực nghiệm trên 10 tập dữ liệu không cân bằng được lấy từ nguồn UCI (Asuncion & Newman, 2007) mô tả trong bảng 1. Nếu tập dữ liệu có sẵn tập học và tập kiểm tra, chúng tôi dùng tập học để xây dựng mô hình và sau đó phân lớp tập kiểm tra bằng mô hình thu được kết quả phân lớp. Nếu tập dữ liệu chưa có sẵn tập học và tập kiểm tra thì chúng tôi sử dụng nghi thức Hold-out để đánh giá hiệu quả. Nghi thức Hold-out thực hiện lấy ngẫu nhiên 2/3 số phần tử từ tập dữ liệu để làm tập học và 1/3 còn lại của tập dữ liệu dùng cho kiểm tra, quá trình này có thể lặp lại k lần ($k=3$ trong thực nghiệm của chúng tôi) và sau đó tính giá trị trung bình trên k kết quả sinh ra làm kết quả cuối cùng.

Để thấy rõ hiệu quả của giải thuật RB Bagging cải tiến mà chúng tôi đề xuất so với giải thuật RB Bagging gốc, chúng tôi tiến hành so sánh kết quả dựa trên các tiêu chí như *precision, recall, accuracy và F1–measure* (van Rijsbergen, 1979). Trong đó p*recision* của một lớp là số phần tử được phân lớp đúng về lớp này chia cho tổng số phần tử được phân về lớp này. *Recall* của một lớp là số phần tử được phân lớp đúng về lớp này chia cho tổng số phần tử của lớp. *Accuracy* là số phần tử được phân lớp đúng của tất cả các lớp chia cho tổng số phần tử. *F1-measure* là trung bình điều hòa của *precision* và *recall*.

Bảng 1: Các tập dữ liệu không cân bằng.

ID	Tập dữ liệu	Số phần tử	Số thuộc tính	Nghi thức	Tỷ lệ lớp nhỏ
1	Letter-A	20 000	16	Trn- tst	3.95%
2	20news	20 000	201	Trn- tst	5%
3	Pendigits	10 992	16	Trn- tst	9.6%
4	Sat-images	6 435	36	Trn- tst	9.73%
5	Adult	48 844	104	Trn- tst	23.9%
6	connect-4	67 557	42	Trn- tst	24.6%
7	Pima	770	8	Hold -out	34.9%
8	Segment	2 310	19	Hold -out	14.3%
9	German	960	24	Hold -out	28.6%
10	Yeast	1 480	8	Hold -out	31.2%

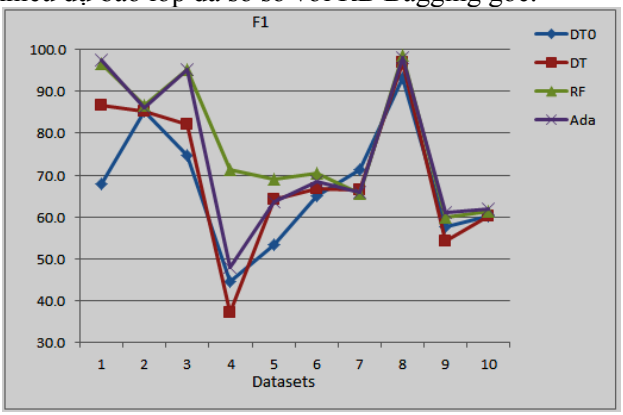
Khi thực thi các giải thuật theo đúng nghi thức kiểm tra được mô tả như trên, chúng tôi thu được kết quả trong bảng 2. Trong đó, cột **DT0** là kết quả thu được khi chạy giải thuật RB Bagging gốc với việc dùng cây quyết định C4.5 như là mô hình học cơ sở, cột **DT**, **RF** và **Ada** là kết quả khi chạy với giải thuật cải tiến

tương ứng với ba mô hình học cơ sở dùng cây quyết định C4.5, rừng ngẫu nhiên và AdaBoost.M1. Khi thực thi, các giải thuật RB Bagging gốc và cải tiến đều xây dựng **200** mô hình học cơ sở. Hơn nữa các mô hình học cơ sở như rừng ngẫu nhiên và AdaBoost.M1 xây dựng **100** cây cho mỗi lần học. Các kết quả tốt nhất được tô đậm và tốt nhì được gạch dưới.

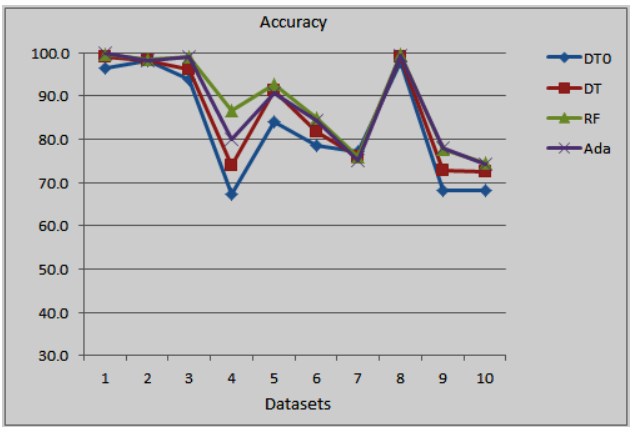
Bảng 2: Kết quả phân lớp của RB Bagging gốc và RB Bagging cải tiến

ID	Precision (%)				Recall (%)				F1-measure (%)				Accuracy (%)			
	DT0	DT	RF	Ada	DT0	DT	RF	Ada	DT0	DT	RF	Ada	DT0	DT	RF	Ada
1	54.5	88.8	96.7	**97.0**	92.5	84.8	96.3	**97.8**	68.0	86.7	96.5	**97.4**	96.3	99.0	99.7	**99.8**
2	74.3	74.3	**76.4**	75.4	**100.0**	**100.0**	99.7	**100.0**	85.3	85.3	**86.5**	86.0	98.2	98.2	**98.4**	98.3
3	62.1	74.0	94.0	**94.7**	93.8	92.3	**96.4**	95.8	74.7	82.1	95.2	**95.3**	93.9	96.1	99.0	**99.1**
4	38.2	45.1	**75.1**	65.2	54.1	31.9	**67.9**	37.8	44.7	37.3	**71.2**	47.9	67.4	73.9	**86.7**	79.9
5	38.6	56.7	**63.2**	54.6	**86.7**	73.9	75.8	76.0	53.4	64.2	**69.0**	63.6	84.0	91.3	**92.8**	90.8
6	52.9	58.6	**66.6**	65.6	**84.3**	77.1	75.2	71.1	65.0	66.6	**70.4**	68.5	78.6	81.7	**85.0**	84.4
7	63.0	63.7	**64.3**	62.6	**80.8**	69.2	67.1	69.8	71.3	**66.3**	65.6	66.0	**77.0**	76.0	75.9	75.0
8	89.5	97.2	**98.7**	97.4	97.5	96.2	98.3	**98.5**	93.3	96.8	**98.5**	97.9	98.0	99.1	**99.6**	99.4
9	47.0	52.2	60.3	**61.0**	**74.2**	56.7	59.3	60.5	57.7	54.3	**59.8**	61.0	68.3	72.7	77.8	**77.9**
10	48.8	55.4	**59.4**	57.1	**78.0**	66.0	63.3	67.6	60.1	60.2	61.2	**61.9**	68.2	72.6	**74.5**	74.1

Từ bảng kết quả phân lớp thu được khi xử lý 10 tập dữ liệu cho thấy giải thuật RB Bagging cải tiến mà chúng tôi đề xuất cho kết quả tốt hơn so với RB Bagging gốc của (Hido & Kashima, 2008). Xét tiêu chí *precision,* thì RB Bagging cải tiến với mô hình cơ sở là rừng ngẫu nhiên và AdaBoost.M1 thắng tất cả 10 tập. Dựa trên tiêu chí *recall*, thì RB Bagging cải tiến vẫn cho kết quả so sánh được với RB Bagging gốc (thắng 5 trên 10 tập). Với tiêu chí *F1* và *accuracy*, RB Bagging cải tiến thắng tất cả 10 tập. Điều này lý giải cho việc thay đổi cách lấy mẫu và mô hình cơ sở trong RB Bagging cải tiến vẫn có được dự báo lớp thiểu số tốt nhưng không làm mất nhiều dự báo lớp đa số so với RB Bagging gốc.



Hình 1: Đồ thị so sánh tiêu chí F1 của các giải thuật trên 10 tập dữ liệu



Hình 2: Đồ thị so sánh tiêu chí Accuracy của các giải thuật trên 10 tập dữ liệu

Quan sát các đồ thị biểu diễn các tiêu chí *F1* (hình 1) và *accuracy* (hình 2) thu được của các giải thuật RB Bagging gốc và RB Bagging cải tiến khi phân lớp 10 tập dữ liệu không cân bằng trong thực nghiệm. Chúng ta nhận thấy rằng đường **DT0** của RB Bagging gốc luôn nằm cận dưới của các đường **DT**, **RF**, **Ada** của RB Bagging cải tiến. Điều này chứng minh rằng giải thuật RB Bagging gốc cho hiệu quả phân loại bị lệch mạnh về lớp thiểu số và làm giảm hiệu quả dự báo của lớp đa số trong khi RB Bagging cải tiến thì vẫn cho kết quả dự báo tốt cho lớp thiểu số nhưng không làm mất hiệu quả dự báo lớp đa số.

Qua kết quả đạt được, chúng tôi tin rằng giải thuật RB Bagging cải tiến mà chúng tôi đề xuất có thể xử lý tốt cho vấn đề phân lớp dữ liệu không cân bằng.

4 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi vừa trình bày giải thuật RB Bagging cải tiến cho phân lớp tập dữ liệu không cân bằng. Ý tưởng mà chúng tôi đề xuất tận dụng được các ưu điểm của RB Bagging gốc (Hido & Kashima, 2008) cải tiến dự báo lớp thiểu số và khắc phục được yếu điểm làm giảm dự báo lớp đa số. RB Bagging cải tiến tập trung cải thiện hiệu quả dự báo cho lớp thiểu số, bên cạnh đó cố gắng đảm bảo sử dụng hầu hết thông tin cho lớp đa số. Để đạt được mục tiêu này, chúng tôi đề xuất hai cải tiến: điều chỉnh cách lấy mẫu giảm phần tử lớp đa số theo hàm phân phối nhị thức âm ở mỗi lần của RB Bagging để không làm mất quá nhiều dự báo lớp đa số, thay thế mô hình học cơ sở là cây quyết định C4.5 (Quinlan, 1993) dùng trong RB Bagging bằng giải thuật tập hợp mô hình mạnh hơn như rừng ngẫu nhiên (Breiman, 2001) và AdaBoost (Freund & Schapire, 1995). Kết quả thực nghiệm trên 10 tập dữ liệu không cân bằng từ kho dữ liệu UCI (Asuncion & Newman, 2007) cho thấy rằng RB Bagging cải tiến cho hiệu quả cao hơn khi so sánh với giải RB Bagging gốc, dựa trên các tiêu chí về precision, recall, F1-measure và accuracy (van Rijsbergen, 1979).

Trong thời gian tới, chúng tôi sẽ thực hiện so sánh hiệu quả giải thuật RB Bagging cải tiến với các giải thuật khác như SmoteBoost (Chawla *et al*., 2003), MetaCost (Domingos, 1999) trong vấn đề phân lớp dữ liệu không cân bằng.

TÀI LIỆU THAM KHẢO

Asuncion, A. & Newman, D.J.: UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science, 2007. [http://www.ics.uci.edu/~m-learn/MLRepository.html]

Breiman, L., Friedman, J., Olshen, R. and Stone C.: *Classification and Regression Trees*. Chapman & Hall, New York, 1984.

Breiman, L.: Bagging predictors. *Machine Learning* 24(2):123–140, 1996.

Breiman, L.: Random Forests. *Machine Learning*, 45(1):5-32, 2001.

Chawla, N., Japkowicz, N. and Kolcz, A.: *ICML'Workshop on Learning from Imbalanced Data Sets*. 2003.

Chawla, N., Japkowicz, N. and Kolcz, A.: Special Issue on Class Imbalances. In *SIGKDD Explorations* Vol. 6, 2004.

Chawla, N., Lazarevic, A., Hall, L.O. and Bowyer, K.W.: SMOTEBoost: Improving prediction of the minority class in boosting. In proc. of European Conf. on Principles and Practice of Knowledge Discovery in Databases, pp. 107–119, 2003.

Domingos, P.: Metacost: A general method for making classifiers cost sensitive. In proc. of Intl Conf. on Knowledge Discovery and Data Mining, pp. 155–164, 1999.

Freund, Y. and Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. In Computational Learning Theory: Proceedings of the Second European Conference, pp. 23–37, 1995.

Hido, S. and Kashima, H.: Roughly balanced bagging for imbalanced data. In proc. of SIAM Intl Conference on Data Mining, pp. 143–152, 2008.

Ihaka, R. and Gentleman, R.: R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299-314, 1996.

Lenca, P., Lallich, S., Do, T-N. and Pham, N-K.: A comparison of different off-centered entropies to deal with class imbalance for decision trees. In The Pacific-Asia Conference on Knowledge Discovery and Data Mining, LNAI 5012, pp. 634–643, 2008.

Liu, X.-Y., Wu, J. and Zhou, Z.-H.: *Exploratory under-sampling for class-imbalance learning*. In proc. of Sixth IEEE Intl Conf. on Data Mining (ICDM'06), pp. 965–969, 2006.

Liu, X-Y. and Zhou, Z-H.: The influence of class imbalance on costsensitive learning: An empirical study. In proc. of Sixth IEEE Intl Conf. on Data Mining (ICDM'06), pp. 970–974, 2006.

Quinlan, J.: *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, 1993.

van Rijsbergen, C.V.: *Information Retrieval. Butterworth*, 1979.

Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

Weiss, G.M. and Provost, F.: Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research* Vol.(19):315–354, 2003.

Yang, Q. and Wu, X.: 10 *Challenging Problems in Data Mining Research*. Intl Journal of Information Technology and Decision Making 5(4), 597–604, 2006.