
Learning Imbalanced Data with Random Forests

Chao Chen (Stat., UC Berkeley)

chenchao@stat. Berkeley. EDU

Andy Liaw (Merck Research Labs)

andy_liaw@merck.com

Leo Breiman (Stat., UC Berkeley)

leo@stat. Berkeley. EDU

Interface 2004, Baltimore

Outline

- Imbalanced data
- Common approaches and recent works
- “Balanced” random forests
- “Weighted” random forests
- Some comparisons
- Conclusion

Imbalanced Data

- Data for many classification problems are inherently **imbalanced**
 - One large, “normal” class (negative) and one small/rare, “interesting” class (positive)
 - E.g.: rare diseases, fraud detection, compound screening in drug discovery, etc.
- Why is this a problem?
 - Most machine learning algorithms focus on overall accuracy, and “break down” with moderate imbalance in the data
 - Even some cost-sensitive algorithms don’t work well when imbalance is extreme

Common Approaches

- Up-sampling minority class
 - random sampling with replacement
 - strategically add cases that reduce error
- Down-sampling majority class
 - random sampling
 - strategically omit cases that do not help
- Cost-sensitive learning
 - build misclassification cost into the algorithm
- Down-sampling tends to work better empirically, but loses some information, as not all training data are used

Recent Work

- One-sided sampling
- SMOTE: Synthetic Minority Oversampling TEchnique (Chawla et al, 2002)
- SMOTEBoost
- SHRINK

Random Forest

- A supervised learning algorithm, constructed by combining multiple decision trees (Breiman, 2001)
- Draw a bootstrap sample of the data
- Grow an **un-pruned** tree
 - At each node, only a **small, random subset** of predictor variables are tried to split that node
- Repeat as many times as you'd like
- Make predictions using all trees

“Balanced” Random Forest

- Natural integration of down-sampling majority class and ensemble learning
- For each tree in RF, down-sample the majority class to the same size as the minority class
- Given enough trees, all training data are used, so no loss of information
- Computationally efficient, since each tree only sees a small sample

“Weighted” Random Forest

- Incorporate class weights in several places of the RF algorithm:
 - Weighted Gini for split selection
 - Class-weighted votes at terminal nodes for node class
 - Weighted votes over all trees, using average weights at terminal nodes
- Using weighted Gini alone isn't sufficient

Performance Assessment

Confusion Matrix

	Predicted Positive	Predicted Negative
Positive	True Positive	False Negative
Negative	False Positive	True Negative

- True Positive Rate (TPR): $TP / (TP + FN)$
- True Negative Rate (TNR): $TN / (TN + FP)$
- Precision: $TP / (TP + FP)$
- Recall: same as TPR
- g-mean: $(TPR \times TNR)^{1/2}$
- F-measure: $(2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$

Benchmark Data

Dataset	No. of Var.	No. of Obs.	% Minority
Oil Spill	50	937	4.4
Mammograph	6	11183	2.3
SatImage	36	6435	9.7

Oil Spill Data

Method	TPR	TNR	Precisio n	G-mean	F- meas
1-sided sampling	76.0	86.6	20.5	81.13	32.3
SHRINK	82.5	60.9	8.85	70.9	16.0
SMOTE	89.5	78.9	16.4	84.0	27.7
BRF	73.2	91.6	28.6	81.9	41.1
WRF	92.7	82.4	19.4	87.4	32.1

Performance for 1-sided sampling, SHRINK, and SMOTE taken from Chawla, et al (2002).

Mammography Data

Method	TPR	TNR	Precisio n	G- mean	F- meas
RIPPER	48.1	99.6	74.7	69.2	58.1
SMOTE	62.2	99.0	60.5	78.5	60.4
SMOTE-Boost	62.6	99.5	74.5	78.9	68.1
BRF	76.5	98.2	50.5	86.7	60.8
WRF	72.7	99.2	69.7	84.9	71.1

Performance for RIPPER, SMOTE, and SMOTE-Boost taken from Chawla, et al (2003).

Satimage Data

Method	TPR	TNR	Precisio n	G- mean	F- meas
RIPPER	47.4	97.6	67.9	68.0	55.5
SMOTE	74.9	91.3	48.1	82.7	58.3
SMOTE-Boost	67.9	97.2	72.7	81.2	70.2
BRF	77.0	93.6	56.3	84.9	65.0
WRF	77.5	94.6	60.5	85.6	68.0

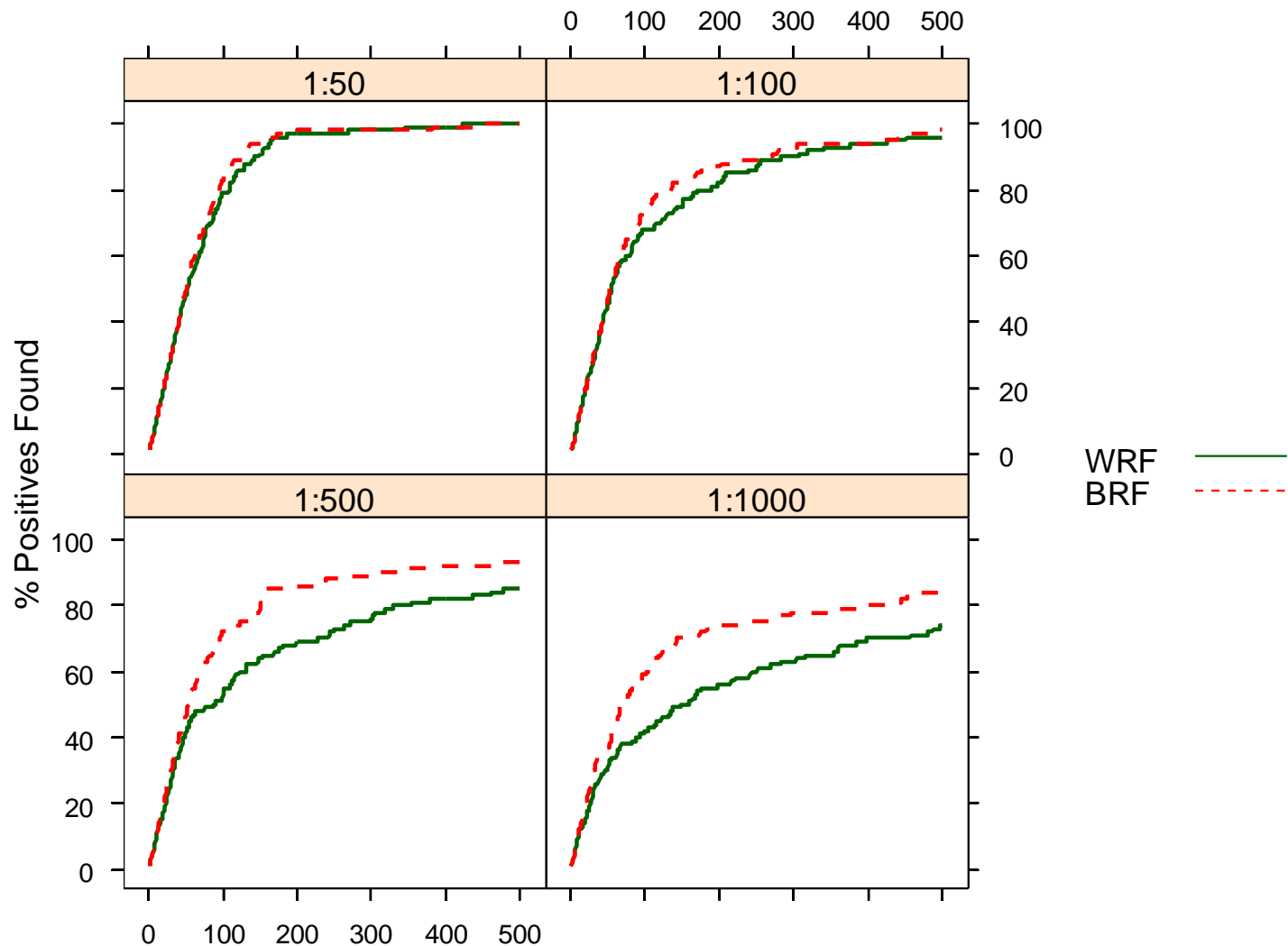
Performance for RIPPER, SMOTE, and SMOTE-Boost taken from Chawla, et al (2003).

A Simple Experiment:

2Norm

- Fix size of one class at 100, vary the size of other class among $5e3$, $1e4$, $5e4$ and $1e5$
- Train both WRF and BRF, predict on same size test set
 - WRF: use reciprocal of class ratio as weights
 - BRF: draw 100 from each class w/replacement to grow each tree
- With usual prediction, BRF has better false negative rate; WRF has better true positive rate
- Compare cumulative gain to see difference

Comparing Cumulative Gain



To Wrap Up...

- We propose two methods of learning imbalanced data with random forests
 - BRF: down-sampling majority in each tree
 - WRF: incorporate class weights in several places
- Both show improvements over existing methods
- The two are about equally effective on real; hard to pick a winner
- Need further study to see if/when/why one works better than the other

Free Software

- Random Forest (Breiman & Cutler): Fortran code, implements WRF, available at <http://stat-www.berkeley.edu/users/breiman/RandomForests/>
- randomForest (Liaw & Wiener): add-on package for R (based on the Fortran code above), implements BRF, available on CRAN
(e.g.: <http://cran.us.r-project.org/src/contrib/PACKAGES.html>)

Acknowledgment

- Adele Cutler (Utah State)
- Vladimir Svetnik, Chris Tong, Ting Wang (BR)
- Matt Wiener (ACSM)