Boosting (machine learning)

In machine learning, **boosting** is an ensemble meta-algorithm for primarily reducing bias, and also variance[1] in supervised learning, and a family of machine learning algorithms that convert weak learners to strong ones.[2] Boosting is based on the question posed by Kearns and Valiant (1988, 1989):[3][4] "Can a set of weak learners create a single strong learner?" A weak learner is defined to be a classifier that is only slightly correlated with the true classification (it can label examples better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification.

Robert Schapire's affirmative answer in a 1990 paper[5] to the question of Kearns and Valiant has had significant ramifications in machine learning and statistics, most notably leading to the development of boosting.[6]

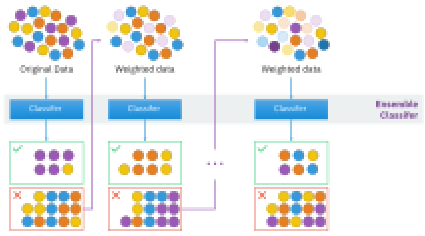
When first introduced, the *hypothesis boosting problem* simply referred to the process of turning a weak learner into a strong learner. "Informally, [the hypothesis boosting] problem asks whether an efficient learning algorithm […] that outputs a hypothesis whose performance is only slightly better than random guessing [i.e. a weak learner] implies the existence of an efficient algorithm that outputs a hypothesis of arbitrary accuracy [i.e. a strong learner]."[3] Algorithms that achieve hypothesis boosting quickly became simply known as "boosting". Freund and Schapire's arcing (Adapt[at]ive Resampling and Combining),[7] as a general technique, is more or less synonymous with boosting.[8]

Contents

Boosting algorithms

While boosting is not algorithmically constrained, most boosting algorithms consist of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier. When they are added, they are weighted in a way that is related to the weak learners' accuracy. After a weak learner is added, the data weights are readjusted, known as "re-weighting". Misclassified input data gain a higher weight and examples that are classified correctly lose weight.[note 1] Thus, future weak learners focus more on the examples that previous weak learners misclassified.

There are many boosting algorithms. The original ones, proposed by Robert Schapire (a recursive majority gate formulation)[5] and Yoav Freund (boost by majority),[9] were not adaptive and could not take full advantage of the weak learners. Schapire and Freund then developed AdaBoost, an adaptive boosting algorithm that won the prestigious Gödel Prize.



An illustration presenting the intuition behind the boosting algorithm, consisting of the parallel learners and weighted dataset.

Only algorithms that are provable boosting algorithms in the probably approximately correct learning formulation can accurately be called *boosting algorithms*. Other algorithms that are similar in spirit to boosting algorithms are sometimes called "leveraging algorithms", although they are also sometimes incorrectly called boosting algorithms.[9]

The main variation between many boosting algorithms is their method of weighting training data points and hypotheses. AdaBoost is very popular and the most significant historically as it was the first algorithm that could adapt to the weak learners. It is often the basis of introductory coverage of boosting in university machine learning courses.[10] There are many more recent algorithms such as LPBoost, TotalBoost, BrownBoost, xgboost, MadaBoost, LogitBoost, and others. Many boosting algorithms fit into the AnyBoost framework,[9] which shows that boosting performs gradient descent in a function space using a convex cost function.

Object categorization in computer vision

Given images containing various known objects in the world, a classifier can be learned from them to automatically classify the objects in future images. Simple classifiers built based on some image feature of the object tend to be weak in categorization performance. Using boosting methods for object categorization is a way to unify the weak classifiers in a special way to boost the overall ability of categorization.

Problem of object categorization

Object categorization is a typical task of computer vision that involves determining whether or not an image contains some specific category of object. The idea is closely related with recognition, identification, and detection. Appearance based object categorization typically contains feature extraction, learning a classifier, and applying the classifier to new examples. There are many ways to represent a category of objects, e.g. from shape analysis, bag of words models, or local descriptors such as SIFT, etc. Examples of supervised classifiers are Naive Bayes classifiers, support vector machines, mixtures of Gaussians, and neural networks. However, research has shown that object categories and their locations in images can be discovered in an unsupervised manner as well.[11]

Status quo for object categorization

The recognition of object categories in images is a challenging problem in computer vision, especially when the number of categories is large. This is due to high intra class variability and the need for generalization across variations of objects within the same category. Objects within one category may look quite different. Even the same object may appear unalike under different viewpoint, scale, and illumination. Background clutter and partial occlusion add difficulties to recognition as well.[12] Humans are able to recognize thousands of object types, whereas most of the existing object recognition systems are trained to recognize only a few, e.g. human faces, cars, simple objects, etc.[13] Research has been very active on dealing with more categories and enabling incremental additions of new categories, and although the general problem remains unsolved, several multi-category objects detectors (for up to hundreds or thousands of categories[14]) have been developed. One means is by feature sharing and boosting.

Boosting for binary categorization

AdaBoost can be used for face detection as an example of binary categorization. The two categories are faces versus background. The general algorithm is as follows:

1. Form a large set of simple features
2. Initialize weights for training images
3. For T rounds
    1. Normalize the weights
    2. For available features from the set, train a classifier using a single feature and evaluate the training error
    3. Choose the classifier with the lowest error
    4. Update the weights of the training images: increase if classified wrongly by this classifier, decrease if correctly
4. Form the final strong classifier as the linear combination of the T classifiers (coefficient larger if training error is small)

After boosting, a classifier constructed from 200 features could yield a 95% detection rate under a 10^{-5} false positive rate.[15]

Another application of boosting for binary categorization is a system that detects pedestrians using patterns of motion and appearance.[16] This work is the first to combine both motion information and appearance information as features to detect a walking person. It takes a similar approach to the Viola-Jones object detection framework.

Boosting for multi-class categorization

Compared with binary categorization, multi-class categorization looks for common features that can be shared across the categories at the same time. They turn to be more generic edge like features. During learning, the detectors for each category can be trained jointly. Compared with training separately, it generalizes better, needs less training data, and requires fewer features to achieve the same performance.

The main flow of the algorithm is similar to the binary case. What is different is that a measure of the joint training error shall be defined in advance. During each iteration the algorithm chooses a classifier of a single feature (features that can be shared by more categories shall be encouraged). This can be done via converting multi-class classification into a binary one (a set of categories versus the rest),[17] or by introducing a penalty error from the categories that do not have the feature of the classifier.[18]

In the paper "Sharing visual features for multiclass and multiview object detection", A. Torralba et al. used GentleBoost for boosting and showed that when training data is limited, learning via sharing features does a much better job than no sharing, given same boosting rounds. Also, for a given performance level, the total number of features required (and therefore the run time cost of the classifier) for the feature sharing detectors, is observed to scale approximately logarithmically with the number of class, i.e., slower than linear growth in the non-sharing case. Similar results are shown in the paper "Incremental learning of object detectors using a visual shape alphabet", yet the authors used AdaBoost for boosting.

Convex vs non-convex boosting algorithms

Boosting algorithms can be based on convex or non-convex optimization algorithms. Convex algorithms, such as AdaBoost and LogitBoost, can be "defeated" by random noise such that they can't learn basic and learnable combinations of weak hypotheses.[19][20] This limitation was pointed out by Long & Servedio in 2008. However, by 2009, multiple authors demonstrated that boosting algorithms based on non-convex optimization, such as BrownBoost, can learn from noisy datasets and can specifically learn the underlying classifier of the Long–Servedio dataset.

See also

- AdaBoost
- Random forest
- Alternating decision tree
- Bootstrap aggregating (bagging)
- Cascading
- BrownBoost
- CoBoosting
- LPBoost
- Logistic regression
- Maximum entropy methods
- Neural networks
- Support vector machines
- Gradient boosting
- Margin classifiers
- Cross-validation
- Machine learning
- List of datasets for machine learning research

Implementations

- Scikit-learn, an open source machine learning library for python
- Orange, a free data mining software suite, module Orange.ensemble (http://docs.orange.biolab.si/reference/rst/Orange.ensemble.html)
- Weka is a machine learning set of tools that offers variate implementations of boosting algorithms like AdaBoost and LogitBoost
- R package GBM (https://cran.r-project.org/web/packages/gbm/index.html) (Generalized Boosted Regression Models) implements extensions to Freund and Schapire's AdaBoost algorithm and Friedman's gradient boosting machine.
- jboost (https://sourceforge.net/projects/jboost/); AdaBoost, LogitBoost, RobustBoost, Boostexter and alternating decision trees
- R package adabag (https://cran.r-project.org/web/packages/adabag/index.html): Applies Multiclass AdaBoost.M1, AdaBoost-SAMME and Bagging

- R package xgboost (https://cran.r-project.org/web/packages/xgboost/index.html): An implementation of gradient boosting for linear and tree-based models.

Notes

1. Some boosting-based classification algorithms actually decrease the weight of repeatedly misclassified examples; for example boost by majority and BrownBoost.

References

1. Leo Breiman (1996). "BIAS, VARIANCE, AND ARCING CLASSIFIERS" (https://web.archive.org/web/20150119081741/http://oz.berkeley.edu/~breiman/arcall96.pdf) (PDF). TECHNICAL REPORT. Archived from the original (http://oz.berkeley.edu/~breiman/arcall96.pdf) (PDF) on 2015-01-19. Retrieved 19 January 2015. "Arcing [Boosting] is more successful than bagging in variance reduction"
2. Zhou Zhi-Hua (2012). *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC. p. 23. ISBN 978-1439830031. "The term boosting refers to a family of algorithms that are able to convert weak learners to strong learners"
3. Michael Kearns(1988); *Thoughts on Hypothesis Boosting* (http://www.cis.upenn.edu/~mkearns/papers/boostnote.pdf), Unpublished manuscript (Machine Learning class project, December 1988)
4. Michael Kearns; Leslie Valiant (1989). *Crytographic [sic] limitations on learning Boolean formulae and finite automata*. *Symposium on Theory of Computing*. **21**. ACM. pp. 433–444. doi:10.1145/73007.73049 (https://doi.org/10.1145%2F73007.73049). ISBN 978-0897913072. S2CID 536357 (https://api.semanticscholar.org/CorpusID:536357).
5. Schapire, Robert E. (1990). "The Strength of Weak Learnability" (https://web.archive.org/web/20121010030839/http://www.cs.princeton.edu/~schapire/papers/strengthofweak.pdf) (PDF). *Machine Learning*. **5** (2): 197–227. CiteSeerX 10.1.1.20.723 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.723). doi:10.1007/bf00116037 (https://doi.org/10.1007%2Fbf00116037). S2CID 53304535 (https://api.semanticscholar.org/CorpusID:53304535). Archived from the original (http://www.cs.princeton.edu/~schapire/papers/strengthofweak.pdf) (PDF) on 2012-10-10. Retrieved 2012-08-23.
6. Leo Breiman (1998). "Arcing classifier (with discussion and a rejoinder by the author)" (https://doi.org/10.1214%2Faos%2F1024691079). *Ann. Stat.* **26** (3): 801–849. doi:10.1214/aos/1024691079 (https://doi.org/10.1214%2Faos%2F1024691079). "Schapire (1990) proved that boosting is possible. (Page 823)"
7. Yoav Freund and Robert E. Schapire (1997); *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting* (https://www.cis.upenn.edu/~mkearns/teaching/COLT/adaboost.pdf), Journal of Computer and System Sciences, 55(1):119-139
8. Leo Breiman (1998); *Arcing Classifier (with Discussion and a Rejoinder by the Author)* (http://projecteuclid.org/DPubS?service=UI&version=1.0&verb=Display&handle=euclid.aos/1024691079), Annals of Statistics, vol. 26, no. 3, pp. 801-849: "The concept of weak learning was introduced by Kearns and Valiant (1988, 1989), who left open the question of whether weak and strong learnability are equivalent. The question was termed the *boosting problem* since [a solution must] boost the low accuracy of a weak learner to the high accuracy of a strong learner. Schapire (1990) proved that boosting is possible. A *boosting algorithm* is a method that takes a weak learner and converts it into a strong learner. Freund and Schapire (1997) proved that an algorithm similar to arc-fs is boosting.
9. Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean (2000); *Boosting Algorithms as Gradient Descent*, in S. A. Solla, T. K. Leen, and K.-R. Muller, editors, *Advances in Neural Information Processing Systems* 12, pp. 512-518, MIT Press

10. Emer, Eric. "Boosting (AdaBoost algorithm)" (http://math.mit.edu/~rothvoss/18.304.3PM/Presentations/1-Eric-Boosting304FinalRpdf.pdf) (PDF). *MIT*. Retrieved 2018-10-10.
11. Sivic, Russell, Efros, Freeman & Zisserman, "Discovering objects and their location in images", ICCV 2005
12. A. Opelt, A. Pinz, et al., "Generic Object Recognition with Boosting", IEEE Transactions on PAMI 2006
13. M. Marszalek, "Semantic Hierarchies for Visual Object Recognition", 2007
14. "Large Scale Visual Recognition Challenge" (http://image-net.org/challenges/LSVRC/2017/). December 2017.
15. P. Viola, M. Jones, "Robust Real-time Object Detection", 2001
16. Viola, P.; Jones, M.; Snow, D. (2003). *Detecting Pedestrians Using Patterns of Motion and Appearance* (http://www.merl.com/publications/docs/TR2003-90.pdf) (PDF). ICCV.
17. A. Torralba, K. P. Murphy, et al., "Sharing visual features for multiclass and multiview object detection", IEEE Transactions on PAMI 2006
18. A. Opelt, et al., "Incremental learning of object detectors using a visual shape alphabet", CVPR 2006
19. P. Long and R. Servedio. 25th International Conference on Machine Learning (ICML), 2008, pp. 608--615.
20. Long, Philip M.; Servedio, Rocco A. (March 2010). "Random classification noise defeats all convex potential boosters" (https://www.cs.columbia.edu/~rocco/Public/mlj9.pdf) (PDF). *Machine Learning*. **78** (3): 287–304. doi:10.1007/s10994-009-5165-z (https://doi.org/10.1007%2Fs10994-009-5165-z). S2CID 53861 (https://api.semanticscholar.org/CorpusID:53861). Retrieved 2015-11-17.

Further reading

- Yoav Freund and Robert E. Schapire (1997); *A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting* (https://www.cse.ucsd.edu/~yfreund/papers/adaboost.pdf), Journal of Computer and System Sciences, 55(1):119-139
- Robert E. Schapire and Yoram Singer (1999); *Improved Boosting Algorithms Using Confidence-Rated Predictors* (http://citeseer.ist.psu.edu/schapire99improved.html), Machine Learning, 37(3):297-336

External links

- Robert E. Schapire (2003); *The Boosting Approach to Machine Learning: An Overview* (http://www.cs.princeton.edu/courses/archive/spr08/cos424/readings/Schapire2003.pdf), MSRI (Mathematical Sciences Research Institute) Workshop on Nonlinear Estimation and Classification
- Zhou Zhi-Hua (2014) *Boosting 25 years* (http://www.slideshare.net/hustwj/ccl2014-keynote?qid=dc589369-18c7-4c8a-8f79-938981d2418f), CCL 2014 Keynote.
- Zhou, Zhihua (2008). "On the margin explanation of boosting algorithm" (http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/colt08.pdf) (PDF). *In: Proceedings of the 21st Annual Conference on Learning Theory (COLT'08)*: 479–490.
- Zhou, Zhihua (2013). "On the doubt about margin explanation of boosting" (http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/aij13marginbound.pdf) (PDF). *Artificial Intelligence*. **203**: 1–18. arXiv:1009.3613 (https://arxiv.org/abs/1009.3613). doi:10.1016/j.artint.2013.07.002 (https://doi.org/10.1016%2Fj.artint.2013.07.002). S2CID 2828847 (https://api.semanticscholar.org/CorpusID:2828847).

Retrieved from "https://en.wikipedia.org/w/index.php?title=Boosting_(machine_learning)&oldid=1008669877"