Ensemble learning

In statistics and machine learning, **ensemble methods** use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.[1][2][3] Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.

Contents

- **Overview**
- **Ensemble theory**
- **Ensemble size**
- **Common types of ensembles**
  - Bayes optimal classifier
  - Bootstrap aggregating (bagging)
  - Boosting
  - Bayesian model averaging
  - Bayesian model combination
  - Bucket of models
  - Stacking
- **Implementations in statistics packages**
- **Ensemble learning applications**
  - Remote sensing
    - Land cover mapping
    - Change detection
  - Computer security
    - Distributed denial of service
    - Malware Detection
    - Intrusion detection
  - Face recognition
  - Emotion recognition
  - Fraud detection
  - Financial decision-making
  - Medicine
- **See also**
- **References**
- **Further reading**
- **External links**

Overview

Supervised learning algorithms perform the task of searching through a hypothesis space to find a suitable hypothesis that will make good predictions with a particular problem.[4] Even if the hypothesis space contains hypotheses that are very well-suited for a particular problem, it may be very difficult to find a good one. Ensembles combine multiple hypotheses to form a (hopefully) better hypothesis. The term *ensemble* is usually reserved for methods that generate multiple hypotheses using the same base learner. The broader term of *multiple classifier systems* also covers hybridization of hypotheses that are not induced by the same base learner.

Evaluating the prediction of an ensemble typically requires more computation than evaluating the prediction of a single model. In one sense, ensemble learning may be thought of as a way to compensate for poor learning algorithms by performing a lot of extra computation. On the other hand, the alternative is to do a lot more learning on one non-ensemble system. An ensemble system may be more efficient at improving overall accuracy for the same increase in compute, storage, or communication resources by using that increase on two or more methods, than would have been improved by increasing resource use for a single method. Fast algorithms such as decision trees are commonly used in ensemble methods (for example, random forests), although slower algorithms can benefit from ensemble techniques as well.

By analogy, ensemble techniques have been used also in unsupervised learning scenarios, for example in consensus clustering or in anomaly detection.

Ensemble theory

An ensemble is itself a supervised learning algorithm, because it can be trained and then used to make predictions. The trained ensemble, therefore, represents a single hypothesis. This hypothesis, however, is not necessarily contained within the hypothesis space of the models from which it is built. Thus, ensembles can be shown to have more flexibility in the functions they can represent. This flexibility can, in theory, enable them to over-fit the training data more than a single model would, but in practice, some ensemble techniques (especially bagging) tend to reduce problems related to over-fitting of the training data.

Empirically, ensembles tend to yield better results when there is a significant diversity among the models.[5][6] Many ensemble methods, therefore, seek to promote diversity among the models they combine.[7][8] Although perhaps non-intuitive, more random algorithms (like random decision trees) can be used to produce a stronger ensemble than very deliberate algorithms (like entropy-reducing decision trees).[9] Using a variety of strong learning algorithms, however, has been shown to be more effective than using techniques that attempt to *dumb-down* the models in order to promote diversity.[10]

Ensemble size

While the number of component classifiers of an ensemble has a great impact on the accuracy of prediction, there is a limited number of studies addressing this problem. *A priori* determining of ensemble size and the volume and velocity of big data streams make this even more crucial for online ensemble classifiers. Mostly statistical tests were used for determining the proper number of components. More recently, a theoretical framework suggested that there is an ideal number of component classifiers for an ensemble such that having more or less than this number of classifiers would deteriorate the accuracy. It is called "the law of diminishing returns in ensemble construction." Their theoretical framework shows that using the same number of independent component classifiers as class labels gives the highest accuracy.[11][12]

Common types of ensembles

Bayes optimal classifier

The Bayes optimal classifier is a classification technique. It is an ensemble of all the hypotheses in the hypothesis space. On average, no other ensemble can outperform it.[13] The naive Bayes optimal classifier is a version of this that assumes that the data is conditionally independent on the class and makes the computation more feasible. Each hypothesis is given a vote proportional to the likelihood that the training dataset would be sampled from a system if that hypothesis were true. To facilitate training data of finite size, the vote of each hypothesis is also multiplied by the prior probability of that hypothesis. The Bayes optimal classifier can be expressed with the following equation:

$$y = \underset{c_j \in C}{\operatorname{argmax}} \sum_{h_i \in H} P(c_j|h_i)P(T|h_i)P(h_i)$$

where y is the predicted class, C is the set of all possible classes, H is the hypothesis space, P refers to a *probability*, and T is the training data. As an ensemble, the Bayes optimal classifier represents a hypothesis that is not necessarily in H. The hypothesis represented by the Bayes optimal classifier, however, is the optimal hypothesis in *ensemble space* (the space of all possible ensembles consisting only of hypotheses in H).

This formula can be restated using Bayes' theorem, which says that the posterior is proportional to the likelihood times the prior:

$$P(h_i|T) \propto P(T|h_i)P(h_i)$$

hence,

$$y = \underset{c_j \in C}{\operatorname{argmax}} \sum_{h_i \in H} P(c_j|h_i)P(h_i|T)$$

Bootstrap aggregating (bagging)

Bootstrap aggregating, often abbreviated as *bagging*, involves having each model in the ensemble vote with equal weight. In order to promote model variance, bagging trains each model in the ensemble using a randomly drawn subset of the training set. As an example, the random forest algorithm combines random decision trees with bagging to achieve very high classification accuracy.[14]

In bagging the samples are generated in such a way that the samples are different from each other however replacement is allowed. Replacement means that an instance can occur in multiple samples a multiple times or it can not appear in some samples at all. These samples are then given to multiple learners and then the results from each learner are combined in the form of voting.

Boosting

Boosting involves incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models mis-classified. In some cases, boosting has been shown to yield better accuracy than bagging, but it also tends to be more likely to over-fit the training data. By far, the most common implementation of boosting is Adaboost, although some newer algorithms are reported to achieve better results.

In Boosting, an equal weight (uniform probability distribution) is given to the sample training data (say D1) at the very starting round. This data (D1) is then given to a base learner (say L1). The mis-classified instances by L1 are assigned a weight higher than the correctly classified instances, but keeping in mind that the total probability distribution will be equal to 1. This boosted data (say D2) is then given to second base learner (say L2) and so on. The results are then combined in the form of voting.

Bayesian model averaging

Bayesian model averaging (BMA) makes predictions using an average over several models with weights given by the posterior probability of each model given the data.[15] BMA is known to generally give better answers than a single model, obtained, e.g., via stepwise regression, especially where very different models have nearly identical performance in the training set but may otherwise perform quite differently.

The most obvious question with any technique that uses Bayes' theorem is the prior, i.e., a specification of the probability (subjective, perhaps) that each model is the best to use for a given purpose. Conceptually, BMA can be used with any prior. The ensembleBMA[16] and BMA[17] packages for R use the prior implied by the Bayesian information criterion, (BIC), following Raftery (1995).[18] The BAS package for R supports the use of the priors implied by Akaike information criterion (AIC) and other criteria over the alternative models as well as priors over the coefficients.[19]

The difference between BIC and AIC is the strength of preference for parsimony. The penalty for model complexity is $\ln(n)k$ for the BIC and $2k$ for the AIC. Large sample asymptotic theory has established that if there is a best model then with increasing sample sizes, BIC is strongly consistent, i.e., will almost certainly find it, while AIC may not, because AIC may continue to place excessive posterior probability on models that are more complicated than they need to be. If on the other hand we are more concerned with efficiency, i.e., minimum mean square prediction error, then asymptotically, AIC and AICc are “efficient” while BIC is not.[20]

Burnham and Anderson (1998, 2002) contributed greatly to introducing a wider audience to the basic ideas of Bayesian model averaging and popularizing the methodology.[21] The availability of software, including other free open-source packages for R beyond those mentioned above, helped make the methods accessible to a wider audience.[22]

Haussler et al. (1994) showed that when BMA is used for classification, its expected error is at most twice the expected error of the Bayes optimal classifier.[23]

Bayesian model combination

Bayesian model combination (BMC) is an algorithmic correction to Bayesian model averaging (BMA). Instead of sampling each model in the ensemble individually, it samples from the space of possible ensembles (with model weightings drawn randomly from a Dirichlet distribution having uniform parameters). This modification overcomes the tendency of BMA to converge toward giving all of the weight to a single model. Although BMC is somewhat more computationally expensive than BMA, it tends to yield dramatically better results. The results from BMC have been shown to be better on average (with statistical significance) than BMA, and bagging.[24]

The use of Bayes' law to compute model weights necessitates computing the probability of the data given each model. Typically, none of the models in the ensemble are exactly the distribution from which the training data were generated, so all of them correctly receive a value close to zero for this term. This would work well if the ensemble were big enough to sample the entire model-space, but such is rarely possible. Consequently, each

pattern in the training data will cause the ensemble weight to shift toward the model in the ensemble that is closest to the distribution of the training data. It essentially reduces to an unnecessarily complex method for doing model selection.

The possible weightings for an ensemble can be visualized as lying on a simplex. At each vertex of the simplex, all of the weight is given to a single model in the ensemble. BMA converges toward the vertex that is closest to the distribution of the training data. By contrast, BMC converges toward the point where this distribution projects onto the simplex. In other words, instead of selecting the one model that is closest to the generating distribution, it seeks the combination of models that is closest to the generating distribution.

The results from BMA can often be approximated by using cross-validation to select the best model from a bucket of models. Likewise, the results from BMC may be approximated by using cross-validation to select the best ensemble combination from a random sampling of possible weightings.

Bucket of models

A "bucket of models" is an ensemble technique in which a model selection algorithm is used to choose the best model for each problem. When tested with only one problem, a bucket of models can produce no better results than the best model in the set, but when evaluated across many problems, it will typically produce much better results, on average, than any model in the set.

The most common approach used for model-selection is cross-validation selection (sometimes called a "bake-off contest"). It is described with the following pseudo-code:

```
For each model m in the bucket:
    Do c times: (where 'c' is some constant)
        Randomly divide the training dataset into two datasets: A, and B.
        Train m with A
        Test m with B
Select the model that obtains the highest average score
```

Cross-Validation Selection can be summed up as: "try them all with the training set, and pick the one that works best".[25]

Gating is a generalization of Cross-Validation Selection. It involves training another learning model to decide which of the models in the bucket is best-suited to solve the problem. Often, a perceptron is used for the gating model. It can be used to pick the "best" model, or it can be used to give a linear weight to the predictions from each model in the bucket.

When a bucket of models is used with a large set of problems, it may be desirable to avoid training some of the models that take a long time to train. Landmark learning is a meta-learning approach that seeks to solve this problem. It involves training only the fast (but imprecise) algorithms in the bucket, and then using the performance of these algorithms to help determine which slow (but accurate) algorithm is most likely to do best.[26]

Stacking

Stacking (sometimes called *stacked generalization*) involves training a learning algorithm to combine the predictions of several other learning algorithms. First, all of the other algorithms are trained using the available data, then a combiner algorithm is trained to make a final prediction using all the predictions of the other algorithms as additional inputs. If an arbitrary combiner algorithm is used, then stacking can theoretically represent any of the ensemble techniques described in this article, although, in practice, a logistic regression model is often used as the combiner.

Stacking typically yields performance better than any single one of the trained models.[27] It has been successfully used on both supervised learning tasks (regression,[28] classification and distance learning [29]) and unsupervised learning (density estimation).[30] It has also been used to estimate bagging's error rate.[3][31] It has been reported to out-perform Bayesian model-averaging.[32] The two top-performers in the Netflix competition utilized *blending*, which may be considered to be a form of stacking.[33]

Implementations in statistics packages

- R: at least three packages offer Bayesian model averaging tools,[34] including the BMS (an acronym for Bayesian Model Selection) package,[35] the BAS (an acronym for Bayesian Adaptive Sampling) package,[36] and the BMA package.[37]
- Python: Scikit-learn, a package for machine learning in Python offers packages for ensemble learning including packages for bagging and averaging methods.
- MATLAB: classification ensembles are implemented in Statistics and Machine Learning Toolbox.[38]

Ensemble learning applications

In the recent years, due to the growing computational power which allows training large ensemble learning in a reasonable time frame, the number of its applications has grown increasingly.[39] Some of the applications of ensemble classifiers include:

Remote sensing

Land cover mapping

Land cover mapping is one of the major applications of Earth observation satellite sensors, using remote sensing and geospatial data, to identify the materials and objects which are located on the surface of target areas. Generally, the classes of target materials include roads, buildings, rivers, lakes, and vegetation.[40] Some different ensemble learning approaches based on artificial neural networks,[41] kernel principal component analysis (KPCA),[42] decision trees with boosting,[43] random forest[40] and automatic design of multiple classifier systems,[44] are proposed to efficiently identify land cover objects.

Change detection

Change detection is an image analysis problem, consisting of the identification of places where the land cover has changed over time. Change detection is widely used in fields such as urban growth, forest and vegetation dynamics, land use and disaster monitoring.[45] The earliest applications of ensemble classifiers in change detection are designed with the majority voting,[46] Bayesian average and the maximum posterior probability.[47]

Computer security

Distributed denial of service

Distributed denial of service is one of the most threatening cyber-attacks that may happen to an internet service provider.[39] By combining the output of single classifiers, ensemble classifiers reduce the total error of detecting and discriminating such attacks from legitimate flash crowds.[48]

Malware Detection

Classification of malware codes such as computer viruses, computer worms, trojans, ransomware and spywares with the usage of machine learning techniques, is inspired by the document categorization problem.[49] Ensemble learning systems have shown a proper efficacy in this area.[50][51]

Intrusion detection

An intrusion detection system monitors computer network or computer systems to identify intruder codes like an anomaly detection process. Ensemble learning successfully aids such monitoring systems to reduce their total error.[52][53]

Face recognition

Face recognition, which recently has become one of the most popular research areas of pattern recognition, copes with identification or verification of a person by their digital images.[54]

Hierarchical ensembles based on Gabor Fisher classifier and independent component analysis preprocessing techniques are some of the earliest ensembles employed in this field.[55][56][57]

Emotion recognition

While speech recognition is mainly based on deep learning because most of the industry players in this field like Google, Microsoft and IBM reveal that the core technology of their speech recognition is based on this approach, speech-based emotion recognition can also have a satisfactory performance with ensemble learning.[58][59]

It is also being successfully used in facial emotion recognition.[60][61][62]

Fraud detection

Fraud detection deals with the identification of bank fraud, such as money laundering, credit card fraud and telecommunication fraud, which have vast domains of research and applications of machine learning. Because ensemble learning improves the robustness of the normal behavior modelling, it has been proposed as an efficient technique to detect such fraudulent cases and activities in banking and credit card systems.[63][64]

Financial decision-making

The accuracy of prediction of business failure is a very crucial issue in financial decision-making. Therefore, different ensemble classifiers are proposed to predict financial crises and financial distress.[65] Also, in the trade-based manipulation problem, where traders attempt to manipulate stock prices by buying and selling activities, ensemble classifiers are required to analyze the changes in the stock market data and detect suspicious symptom of stock price manipulation.[65]

Medicine

Ensemble classifiers have been successfully applied in neuroscience, proteomics and medical diagnosis like in neuro-cognitive disorder (i.e. Alzheimer or myotonic dystrophy) detection based on MRI datasets.[66][67][68]

See also

- Ensemble averaging (machine learning)
- Bayesian structural time series (BSTS)

References

1. Opitz, D.; Maclin, R. (1999). "Popular ensemble methods: An empirical study" (https://doi.org/10.1613%2Fjair.614). *Journal of Artificial Intelligence Research*. **11**: 169–198. doi:10.1613/jair.614 (https://doi.org/10.1613%2Fjair.614).
2. Polikar, R. (2006). "Ensemble based systems in decision making". *IEEE Circuits and Systems Magazine*. **6** (3): 21–45. doi:10.1109/MCAS.2006.1688199 (https://doi.org/10.1109%2FMCAS.2006.1688199). S2CID 18032543 (https://api.semanticscholar.org/CorpusID:18032543).
3. Rokach, L. (2010). "Ensemble-based classifiers". *Artificial Intelligence Review*. **33** (1–2): 1–39. doi:10.1007/s10462-009-9124-7 (https://doi.org/10.1007%2Fs10462-009-9124-7). S2CID 11149239 (https://api.semanticscholar.org/CorpusID:11149239).
4. Blockeel H. (2011). "Hypothesis Space". *Encyclopedia of Machine Learning*: 511–513. doi:10.1007/978-0-387-30164-8_373 (https://doi.org/10.1007%2F978-0-387-30164-8_373). ISBN 978-0-387-30768-8.
5. Kuncheva, L. and Whitaker, C., Measures of diversity in classifier ensembles, *Machine Learning*, 51, pp. 181-207, 2003
6. Sollich, P. and Krogh, A., *Learning with ensembles: How overfitting can be useful*, Advances in Neural Information Processing Systems, volume 8, pp. 190-196, 1996.
7. Brown, G. and Wyatt, J. and Harris, R. and Yao, X., Diversity creation methods: a survey and categorisation., *Information Fusion*, 6(1), pp.5-20, 2005.
8. Adeva, J. J. García; Cerviño, Ulises; Calvo, R. "Accuracy and Diversity in Ensembles of Text Categorisers" (https://www.clei.org/cleiej/index.php/cleiej/article/view/319/112) (PDF). *CLEI Journal*. **8** (2): 1–12. doi:10.19153/cleiej.8.2.1 (https://doi.org/10.19153%2Fcleiej.8.2.1).
9. Ho, T., Random Decision Forests, *Proceedings of the Third International Conference on Document Analysis and Recognition*, pp. 278-282, 1995.
10. Gashler, M.; Giraud-Carrier, C.; Martinez, T. (2008). "Decision Tree Ensemble: Small Heterogeneous Is Better Than Large Homogeneous" (http://axon.cs.byu.edu/papers/gashler2008icmla.pdf) (PDF). *The Seventh International Conference on Machine Learning and Applications*. **2008**: 900–905. doi:10.1109/ICMLA.2008.154 (https://doi.org/10.1109%2FICMLA.2008.154). ISBN 978-0-7695-3495-4. S2CID 614810 (https://api.semanticscholar.org/CorpusID:614810).
11. R. Bonab, Hamed; Can, Fazli (2016). *A Theoretical Framework on the Ideal Number of Classifiers for Online Ensembles in Data Streams* (http://dl.acm.org/citation.cfm?id=2983907). CIKM. USA: ACM. p. 2053.
12. R. Bonab, Hamed; Can, Fazli (2019). *Less Is More: A Comprehensive Framework for the Number of Components of Ensemble Classifiers*. TNNLS. USA: IEEE. arXiv:1709.02925 (https://arxiv.org/abs/1709.02925).
13. Tom M. Mitchell, *Machine Learning*, 1997, pp. 175
14. Breiman, L., Bagging Predictors, *Machine Learning*, 24(2), pp.123-140, 1996.

15. e.g., Jennifer A. Hoeting; David Madigan; Adrian Raftery; Chris Volinsky (1999). "Bayesian Model Averaging: A Tutorial" (https://projecteuclid.org/euclid.ss/1009212519). *Statistical Science*. ISSN 0883-4237 (https://www.worldcat.org/issn/0883-4237). Wikidata Q98974344.
16. Chris Fraley; Adrian Raftery; J. McLean Sloughter; Tilmann Gneiting, *ensembleBMA: Probabilistic Forecasting using Ensembles and Bayesian Model Averaging*, Wikidata Q98972500
17. Adrian Raftery; Jennifer Hoeting; Chris Volinsky; Ian Painter; Ka Yee Yeung, *BMA: Bayesian Model Averaging* (https://github.com/hanase/BMA), Wikidata Q91674106.
18. Adrian Raftery (1995). "Bayesian model selection in social research". *Sociological Methodology*: 111–196. ISSN 0081-1750 (https://www.worldcat.org/issn/0081-1750). Wikidata Q91670340.
19. Merlise A. Clyde; Michael L. Littman; Quanli Wang; Joyee Ghosh; Yingbo Li; Don van de Bergh, *BAS: Bayesian Variable Selection and Model Averaging using Bayesian Adaptive Sampling*, Wikidata Q98974089.
20. Gerda Claeskens; Nils Lid Hjort (2008), *Model selection and model averaging*, Cambridge University Press, Wikidata Q62568358, ch. 4.
21. Kenneth P. Burnham; David R. Anderson (1998), *Model Selection and Multimodel Inference: A practical information-theoretic approach*, Wikidata Q62670082 and Kenneth P. Burnham; David R. Anderson (2002), *Model Selection and Multimodel Inference: A practical information-theoretic approach*, Springer Science+Business Media, Wikidata Q76889160.
22. The Wikiversity article on Searching R Packages mentions several ways to find available packages for something like this. For example, “sos::findFn('{Bayesian model averaging}')” from within R will search for help files in contributed packages that includes the search term and open two tabs in the default browser. The first will list all the help files found sorted by package. The second summarizes the packages found, sorted by the apparent strength of the match.
23. Haussler, David; Kearns, Michael; Schapire, Robert E. (1994). "Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension" (https://doi.org/10.1007%2Fbf00993163). *Machine Learning*. **14**: 83–113. doi:10.1007/bf00993163 (https://doi.org/10.1007%2Fbf00993163).
24. Monteith, Kristine; Carroll, James; Seppi, Kevin; Martinez, Tony. (2011). *Turning Bayesian Model Averaging into Bayesian Model Combination* (http://axon.cs.byu.edu/papers/Kristine.ijcnn2011.pdf) (PDF). Proceedings of the International Joint Conference on Neural Networks IJCNN'11. pp. 2657–2663.
25. Saso Dzeroski, Bernard Zenko, *Is Combining Classifiers Better than Selecting the Best One (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.108.6096)*, Machine Learning, 2004, pp. 255-273
26. Bensusan, Hilan; Giraud-Carrier, Christophe (2000). "Discovering Task Neighbourhoods through Landmark Learning Performances" (https://link.springer.com/content/pdf/10.1007/3-540-45372-5_32.pdf) (PDF). *Principles of Data Mining and Knowledge Discovery*. Lecture Notes in Computer Science. **1910**. pp. 325–330. doi:10.1007/3-540-45372-5_32 (https://doi.org/10.1007%2F3-540-45372-5_32). ISBN 978-3-540-41066-9.
27. Wolpert (1992). "Stacked Generalization". *Neural Networks*. **5** (2): 241–259. doi:10.1016/s0893-6080(05)80023-1 (https://doi.org/10.1016%2Fs0893-6080%2805%2980023-1).
28. Breiman, Leo (1996). "Stacked regressions" (https://doi.org/10.1007%2FBF00117832). *Machine Learning*. **24**: 49–64. doi:10.1007/BF00117832 (https://doi.org/10.1007%2FBF00117832).
29. Ozay, M.; Yarman Vural, F. T. (2013). "A New Fuzzy Stacked Generalization Technique and Analysis of its Performance". arXiv:1204.0171 (https://arxiv.org/abs/1204.0171). Bibcode:2012arXiv1204.0171O (https://ui.adsabs.harvard.edu/abs/2012arXiv1204.0171O).

30. Smyth, P. and Wolpert, D. H., *Linearly Combining Density Estimators via Stacking (https://link.springer.com/content/pdf/10.1023/A:1007511322260.pdf)*, Machine Learning Journal, 36, 59-83, 1999
31. Wolpert, D.H., and Macready, W.G., *An Efficient Method to Estimate Bagging's Generalization Error (https://link.springer.com/content/pdf/10.1023/A:1007519102914.pdf)*, Machine Learning Journal, 35, 41-55, 1999
32. Clarke, B., *Bayes model averaging and stacking when model approximation error cannot be ignored*, Journal of Machine Learning Research, pp 683-712, 2003
33. Sill, J.; Takacs, G.; Mackey, L.; Lin, D. (2009). "Feature-Weighted Linear Stacking". arXiv:0911.0460 (https://arxiv.org/abs/0911.0460). Bibcode:2009arXiv0911.0460S (https://ui.adsabs.harvard.edu/abs/2009arXiv0911.0460S).
34. Amini, Shahram M.; Parmeter, Christopher F. (2011). "Bayesian model averaging in R" (https://core.ac.uk/download/pdf/6494889.pdf) (PDF). *Journal of Economic and Social Measurement*. **36** (4): 253–287. doi:10.3233/JEM-2011-0350 (https://doi.org/10.3233%2FJEM-2011-0350).
35. "BMS: Bayesian Model Averaging Library" (https://cran.r-project.org/web/packages/BMS/index.html). *The Comprehensive R Archive Network*. 2015-11-24. Retrieved September 9, 2016.
36. "BAS: Bayesian Model Averaging using Bayesian Adaptive Sampling" (https://cran.r-project.org/web/packages/BAS/index.html). *The Comprehensive R Archive Network*. Retrieved September 9, 2016.
37. "BMA: Bayesian Model Averaging" (https://cran.r-project.org/web/packages/BMA/index.html). *The Comprehensive R Archive Network*. Retrieved September 9, 2016.
38. "Classification Ensembles" (https://uk.mathworks.com/help/stats/classification-ensembles.html). *MATLAB & Simulink*. Retrieved June 8, 2017.
39. Woźniak, Michał; Graña, Manuel; Corchado, Emilio (March 2014). "A survey of multiple classifier systems as hybrid systems". *Information Fusion*. **16**: 3–17. doi:10.1016/j.inffus.2013.04.006 (https://doi.org/10.1016%2Fj.inffus.2013.04.006). hdl:10366/134320 (https://hdl.handle.net/10366%2F134320).
40. Rodriguez-Galiano, V.F.; Ghimire, B.; Rogan, J.; Chica-Olmo, M.; Rigol-Sanchez, J.P. (January 2012). "An assessment of the effectiveness of a random forest classifier for land-cover classification". *ISPRS Journal of Photogrammetry and Remote Sensing*. **67**: 93–104. Bibcode:2012JPRS...67...93R (https://ui.adsabs.harvard.edu/abs/2012JPRS...67...93R). doi:10.1016/j.isprsjprs.2011.11.002 (https://doi.org/10.1016%2Fj.isprsjprs.2011.11.002).
41. Giacinto, Giorgio; Roli, Fabio (August 2001). "Design of effective neural network ensembles for image classification purposes". *Image and Vision Computing*. **19** (9–10): 699–707. CiteSeerX 10.1.1.11.5820 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.5820). doi:10.1016/S0262-8856(01)00045-2 (https://doi.org/10.1016%2FS0262-8856%2801%2900045-2).
42. Xia, Junshi; Yokoya, Naoto; Iwasaki, Yakira (March 2017). *A novel ensemble classifier of hyperspectral and LiDAR data using morphological features*. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 6185–6189. doi:10.1109/ICASSP.2017.7953345 (https://doi.org/10.1109%2FICASSP.2017.7953345). ISBN 978-1-5090-4117-6. S2CID 40210273 (https://api.semanticscholar.org/CorpusID:40210273).
43. Mochizuki, S.; Murakami, T. (November 2012). "Accuracy comparison of land cover mapping using the object-oriented image classification with machine learning algorithms". *33rd Asian Conference on Remote Sensing 2012, ACRS 2012*. **1**: 126–133.
44. Giacinto, G.; Roli, F.; Fumera, G. (September 2000). *Design of effective multiple classifier systems by clustering of classifiers*. *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. **2**. pp. 160–163. CiteSeerX 10.1.1.11.5328 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.5328). doi:10.1109/ICPR.2000.906039 (https://doi.org/10.1109%2FICPR.2000.906039). ISBN 978-0-7695-0750-7. S2CID 2625643 (https://api.semanticscholar.org/CorpusID:2625643).

45. Du, Peijun; Liu, Sicong; Xia, Junshi; Zhao, Yindi (January 2013). "Information fusion techniques for change detection from multi-temporal remote sensing images". *Information Fusion*. **14** (1): 19–27. doi:10.1016/j.inffus.2012.05.003 (https://doi.org/10.1016%2Fj.inffus.2012.05.003).
46. Defined by Bruzzone et al. (2002) as "The data class that receives the largest number of votes is taken as the class of the input pattern", this is *simple majority*, more accurately described as *plurality* voting.
47. Bruzzone, Lorenzo; Cossu, Roberto; Vernazza, Gianni (December 2002). "Combining parametric and non-parametric algorithms for a partially unsupervised classification of multitemporal remote-sensing images" (http://eprints.biblio.unitn.it/105/1/24.pdf) (PDF). *Information Fusion*. **3** (4): 289–297. doi:10.1016/S1566-2535(02)00091-X (https://doi.org/10.1016%2FS1566-2535%2802%2900091-X).
48. Raj Kumar, P. Arun; Selvakumar, S. (July 2011). "Distributed denial of service attack detection using an ensemble of neural classifier". *Computer Communications*. **34** (11): 1328–1341. doi:10.1016/j.comcom.2011.01.012 (https://doi.org/10.1016%2Fj.comcom.2011.01.012).
49. Shabtai, Asaf; Moskovitch, Robert; Elovici, Yuval; Glezer, Chanan (February 2009). "Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey". *Information Security Technical Report*. **14** (1): 16–29. doi:10.1016/j.istr.2009.03.003 (https://doi.org/10.1016%2Fj.istr.2009.03.003).
50. Zhang, Boyun; Yin, Jianping; Hao, Jingbo; Zhang, Dingxing; Wang, Shulin (2007). *Malicious Codes Detection Based on Ensemble Learning*. *Autonomic and Trusted Computing*. Lecture Notes in Computer Science. **4610**. pp. 468–477. doi:10.1007/978-3-540-73547-2_48 (https://doi.org/10.1007%2F978-3-540-73547-2_48). ISBN 978-3-540-73546-5.
51. Menahem, Eitan; Shabtai, Asaf; Rokach, Lior; Elovici, Yuval (February 2009). "Improving malware detection by applying multi-inducer ensemble". *Computational Statistics & Data Analysis*. **53** (4): 1483–1494. CiteSeerX 10.1.1.150.2722 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.150.2722). doi:10.1016/j.csda.2008.10.015 (https://doi.org/10.1016%2Fj.csda.2008.10.015).
52. Locasto, Michael E.; Wang, Ke; Keromytis, Angeles D.; Salvatore, J. Stolfo (2005). *FLIPS: Hybrid Adaptive Intrusion Prevention*. *Recent Advances in Intrusion Detection*. Lecture Notes in Computer Science. **3858**. pp. 82–101. CiteSeerX 10.1.1.60.3798 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.3798). doi:10.1007/11663812_5 (https://doi.org/10.1007%2F11663812_5). ISBN 978-3-540-31778-4.
53. Giacinto, Giorgio; Perdisci, Roberto; Del Rio, Mauro; Roli, Fabio (January 2008). "Intrusion detection in computer networks by a modular ensemble of one-class classifiers". *Information Fusion*. **9** (1): 69–82. CiteSeerX 10.1.1.69.9132 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.69.9132). doi:10.1016/j.inffus.2006.10.002 (https://doi.org/10.1016%2Fj.inffus.2006.10.002).
54. Mu, Xiaoyan; Lu, Jiangfeng; Watta, Paul; Hassoun, Mohamad H. (July 2009). *Weighted voting-based ensemble classifiers with application to human face recognition and voice recognition*. *2009 International Joint Conference on Neural Networks*. pp. 2168–2171. doi:10.1109/IJCNN.2009.5178708 (https://doi.org/10.1109%2FIJCNN.2009.5178708). ISBN 978-1-4244-3548-7. S2CID 18850747 (https://api.semanticscholar.org/CorpusID:18850747).
55. Yu, Su; Shan, Shiguang; Chen, Xilin; Gao, Wen (April 2006). *Hierarchical ensemble of Gabor Fisher classifier for face recognition*. *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. pp. 91–96. doi:10.1109/FGR.2006.64 (https://doi.org/10.1109%2FFGR.2006.64). ISBN 978-0-7695-2503-7. S2CID 1513315 (https://api.semanticscholar.org/CorpusID:1513315).
56. Su, Y.; Shan, S.; Chen, X.; Gao, W. (September 2006). *Patch-based gabor fisher classifier for face recognition*. *Proceedings - International Conference on Pattern Recognition*. **2**. pp. 528–531. doi:10.1109/ICPR.2006.917 (https://doi.org/10.1109%2FICPR.2006.917). ISBN 978-0-7695-2521-1. S2CID 5381806 (https://api.semanticscholar.org/CorpusID:5381806).

57. Liu, Yang; Lin, Yongzheng; Chen, Yuehui (July 2008). *Ensemble Classification Based on ICA for Face Recognition*. *Proceedings - 1st International Congress on Image and Signal Processing, IEEE Conference, CISP 2008*. pp. 144–148. doi:10.1109/CISP.2008.581 (https://doi.org/10.1109%2FCISP.2008.581). ISBN 978-0-7695-3119-9. S2CID 16248842 (https://api.semanticscholar.org/CorpusID:16248842).
58. Rieger, Steven A.; Muraleedharan, Rajani; Ramachandran, Ravi P. (2014). *Speech based emotion recognition using spectral feature extraction and an ensemble of kNN classifiers*. *Proceedings of the 9th International Symposium on Chinese Spoken Language Processing, ISCSLP 2014*. pp. 589–593. doi:10.1109/ISCSLP.2014.6936711 (https://doi.org/10.1109%2FISCSLP.2014.6936711). ISBN 978-1-4799-4219-0. S2CID 31370450 (https://api.semanticscholar.org/CorpusID:31370450).
59. Krajewski, Jarek; Batliner, Anton; Kessel, Silke (October 2010). *Comparing Multiple Classifiers for Speech-Based Detection of Self-Confidence - A Pilot Study*. *2010 20th International Conference on Pattern Recognition*. pp. 3716–3719. doi:10.1109/ICPR.2010.905 (https://doi.org/10.1109%2FICPR.2010.905). ISBN 978-1-4244-7542-1. S2CID 15431610 (https://api.semanticscholar.org/CorpusID:15431610).
60. Rani, P. Ithaya; Muneeswaran, K. (25 May 2016). "Recognize the facial emotion in video sequences using eye and mouth temporal Gabor features". *Multimedia Tools and Applications*. **76** (7): 10017–10040. doi:10.1007/s11042-016-3592-y (https://doi.org/10.1007%2Fs11042-016-3592-y). S2CID 20143585 (https://api.semanticscholar.org/CorpusID:20143585).
61. Rani, P. Ithaya; Muneeswaran, K. (August 2016). "Facial Emotion Recognition Based on Eye and Mouth Regions". *International Journal of Pattern Recognition and Artificial Intelligence*. **30** (7): 1655020. doi:10.1142/S021800141655020X (https://doi.org/10.1142%2FS021800141655020X).
62. Rani, P. Ithaya; Muneeswaran, K (28 March 2018). "Emotion recognition based on facial components" (https://doi.org/10.1007%2Fs12046-018-0801-6). *Sādhanā*. **43** (3). doi:10.1007/s12046-018-0801-6 (https://doi.org/10.1007%2Fs12046-018-0801-6).
63. Louzada, Francisco; Ara, Anderson (October 2012). "Bagging k-dependence probabilistic networks: An alternative powerful fraud detection tool". *Expert Systems with Applications*. **39** (14): 11583–11592. doi:10.1016/j.eswa.2012.04.024 (https://doi.org/10.1016%2Fj.eswa.2012.04.024).
64. Sundarkumar, G. Ganesh; Ravi, Vadlamani (January 2015). "A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance". *Engineering Applications of Artificial Intelligence*. **37**: 368–377. doi:10.1016/j.engappai.2014.09.019 (https://doi.org/10.1016%2Fj.engappai.2014.09.019).
65. Kim, Yoonseong; Sohn, So Young (August 2012). "Stock fraud detection using peer group analysis". *Expert Systems with Applications*. **39** (10): 8986–8992. doi:10.1016/j.eswa.2012.02.025 (https://doi.org/10.1016%2Fj.eswa.2012.02.025).
66. Savio, A.; García-Sebastián, M.T.; Chyzyk, D.; Hernandez, C.; Graña, M.; Sistiaga, A.; López de Munain, A.; Villanúa, J. (August 2011). "Neurocognitive disorder detection based on feature vectors extracted from VBM analysis of structural MRI". *Computers in Biology and Medicine*. **41** (8): 600–610. doi:10.1016/j.compbiomed.2011.05.010 (https://doi.org/10.1016%2Fj.compbiomed.2011.05.010). PMID 21621760 (https://pubmed.ncbi.nlm.nih.gov/21621760).
67. Ayerdi, B.; Savio, A.; Graña, M. (June 2013). *Meta-ensembles of classifiers for Alzheimer's disease detection using independent ROI features*. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Lecture Notes in Computer Science. **7931**. pp. 122–130. doi:10.1007/978-3-642-38622-0_13 (https://doi.org/10.1007%2F978-3-642-38622-0_13). ISBN 978-3-642-38621-3.
68. Gu, Quan; Ding, Yong-Sheng; Zhang, Tong-Liang (April 2015). "An ensemble classifier based prediction of G-protein-coupled receptor classes in low homology". *Neurocomputing*. **154**: 110–118. doi:10.1016/j.neucom.2014.12.013 (https://doi.org/10.1016%2Fj.neucom.2014.12.013).

Further reading

- Zhou Zhihua (2012). *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC. ISBN 978-1-439-83003-1.
- Robert Schapire; Yoav Freund (2012). *Boosting: Foundations and Algorithms*. MIT. ISBN 978-0-262-01718-3.

External links

- Robi Polikar (ed.). "Ensemble learning" (http://www.scholarpedia.org/article/Ensemble_learning). *Scholarpedia*.
- The Waffles (machine learning) toolkit contains implementations of Bagging, Boosting, Bayesian Model Averaging, Bayesian Model Combination, Bucket-of-models, and other ensemble techniques