Can Evolutionary Sampling Improve Bagged Ensembles?

Harsh Nisar NISAR.HARSH@GMAIL.COM
Bhanu Pratap Singh BHANUPRATAP.MNIT@GMAIL.COM

Abstract

Perturb and Combine (P&C) group of methods generate multiple versions of the predictor by perturbing the training set or construction and then combining them into a single predictor (Breiman, 1996b). The motive is to improve the accuracy in unstable classification and regression methods. One of the most well known method in this group is Bagging. Arcing or Adaptive Resampling and Combining methods like AdaBoost are smarter variants of P&C methods. In this extended abstract, we lay the groundwork for a new family of methods under the P&C umbrella, known as Evolutionary Sampling (ES). We employ Evolutionary algorithms to suggest smarter sampling in both the feature space (sub-spaces) as well as training samples. We discuss multiple fitness functions to assess ensembles and empirically compare our performance against randomized sampling of training data and feature subspaces.

1. Introduction

Bagging and various variants of the it have been widely popular and studied extensively in the last two decade (Breiman, 2001; 1996a; 1999a; Ho, 1998). There has been notable work in understanding the theoretical underpinning of bootstrap aggregating and as to what makes it such a powerful method (Domingos, 1997; Büchlmann & Yu, 2002). In traditional bagging, each training example is sampled with replacement and with probability $\frac{1}{N}$. Adaptive Resampling and Combining techniques which modify the probability of each training example being sampled based on heuristics have also been developed and widely used (Freund et al., 1996; Breiman, 1999b).

Error based resampling algorithms which try to set the train-set error to zero, designed bagged ensembles with minimal intersection (Papakonstantinou et al., 2014), diver-

sity and uncorrelated errors (Kuncheva & Whitaker, 2003; Tang et al., 2006), importance sampling (Breiman, 1999a) etc. are some of the areas being studied to improve bagged ensembles. Either there are multiple answers to the question, or the answer changes with each dataset.

Instead of figuring out precisely as to *what* sampling and combination of training sets make a bagged ensemble better, we try to fix the *definition* of better, and allow the bootstrapped training sets to *evolve* themselves in order to align with the definition. We generate multiple sampled candidate training sets for the final ensemble and let them compete, mutate and mate their way to the optimal sampling and combination. Evolutionary computation has been used for selection of different predictors to be part of an ensemble (Gagné et al., 2007). To our best knowledge, genetic algorithms haven't been directly used to evolve bootstrapped samples of the training data.

Table 1. Genetic algorithm's experiment setting

GA parameter	Value
Selection	3-way tournament
Crossover	2-point crossover
Population Size	30
Per-individual mutation rate	0.4
Per-individual crossover rate	0.6
Generations	30
Ensemble size in an individual	10

2. Algorithm

Evolutionary computation techniques evolve a population of solution variables (bootstrapped training sets in our case) to optimize towards a given criteria. The fittest offspring across all the generations is considered as the most optimal solution. In Evolutionary Sampling (ES), we followed a standard Genetic Algorithm. Initially, a population of multiple ensembles is generated by randomly sampling from the training data multiple times. Each ensemble (henceforth referred to as an individual) in a generation is evaluated based on a fitness function. Fit individuals are selected for the next generation. After this, crossover is applied on a fixed percentage of individuals wherein two individuals swap their predictors. Post this, a fixed percentage of individuals unaffected by crossover undergo a random mutation. Randomly selected member datasets from the selected individual have rows/features deleted, replaced or inserted with equal probability.

Table 2. Results comparing performance of first individual (FI) and hall of fame (HOF) on unseen data. Values are averages (standard deviations) over 100 runs. Statistical tests are p-values of paired t-tests on the test mean squared error (mse) compared between FI and HOI. Win(%) is the proportion of runs where HOF had a lower mse than FI on unseen data.

Data Set		Parameters	Sub-sampling			Sub-spacing	
			FEMPO	FEMPT	FEGT	FEMPO	FEGT
Servo		HOF - Mean (Std. Dev.)	0.16 (0.11)	0.2 (0.43)	0.26 (0.23)	0.25(0.29)	0.59(0.52)
Nr. Inputs	4	FI - Mean (Std. Dev.)	0.20 (0.15)	0.2 (0.38)	0.28 (0.32)	0.71(0.63)	0.72(0.49)
#Training	167	Win (%)	64	43	52	94	65
#Test	10%	p-value	0	0.475	0.47	0	0.01
Ozone		HOF - Mean (Std. Dev.)	18.20 (4.77)	18.28 (6.02)	18.79 (5.62)	18.48 (5.38)	19.11(5.98)
Nr. Inputs	8	FI - Mean (Std. Dev.)	18.12 (4.58)	18.15 (5.62)	19.66 (6.55)	20.16 (5.59)	20.27(6.02)
#Training	330	Win (%)	51	52	57	67	62
#Test	10%	p-value	0.71	0.59	0.025	0	0.01
Boston		HOF - Mean (Std. Dev.)	12.49 (6.88)	13.09 (7.09)	12.74 (5.4)	20.78 (11.35)	21.02(12.23)
Nr. Inputs	12	FI - Mean (Std. Dev.)	12.96 (6.76)	13.11 (6.6)	13.95 (6.96)	25.07 (13.41)	27.8(15.06)
#Training	506	Win (%)	63	48	58	75	81
#Test	10%	p-value	0.16	0.93	0	0	0
Abalone		HOF - Mean (Std. Dev.)	4.93 (0.37)	5.025 (0.402)	4.81(0.26)	5.09(0.59)	5.02(0.54)
Nr. Inputs	8	FI - Mean (Std. Dev.)	4.92 (0.35)	5.012 (0.409)	4.86(0.22)	5.49(0.71)	5.4(0.62)
#Training	4177	Win (%)	54	40	72	90	84
#Test	25%	p-value	0.42	0.25	0	0	0

We've used the Python package DEAP (Fortin et al., 2012) to implement ES. GA parameters are shown in table 1. As suggested before, instead of understanding as to what makes a bagged ensemble better, we try to rely on the definition of better and try to evolve our ensemble into the same. The fitness function is what guides the sampling and combination of the different sampled datasets. We propose three fitness functions and then try to analyse their performance.

FEMPO: Fitness Each Model Private Out of Bag. It takes each predictor part of the candidate ensemble and measures their performance on the samples that were left out of it's training bag (Breiman, 1996c). Final fitness of the ensemble is the mean of each model's RMSE.

FEMPT: Fitness Each Model Private Test. It is the average of the performance of member predictors on a private test-set which is held out for each sampled dataset during its instantiation.

FEGT: Fitness Ensemble Global Test. During the start of the algorithm, 20% of the training data is set aside. Each ensembles prediction is based on the average prediction of it's member predictors. RMSE is calculated against the set aside global test.

3. Experiment

We conduct experiments on two variants of sampling :- Sub-sampling and Sub-spacing. Sub-sampling works on sampling training examples, whereas sub-spacing works on generating multiple feature sets. We conduct our experiments on 4 benchmark datasets [see table 2]. We compare the mean squared error of the first individual (FI) of the first generation with the hall of fame (best individual) after 30 generations. We assume that the first individual in the

first generation is representative of an ensemble which randomly samples its rows or features like in traditional bagging. We try to analyse whether ES is able to evolve better ensembles starting from random specimens. We uniformly use an unpruned Decision Tree Regressor with max depth arbitrarily set as 5.

4. Results and conclusion

In sub-sampling, FEMPO and FEGT perform significantly better than their random counterparts. Though the win percentages are almost half in many cases, it could be that the algorithm was initialized with an optimal combination and sampling of datasets and there wasn't much scope of improvement. In sub-spacing, both FEMPO and FEGT do significantly better than random sub-spacing. GA has definitely helped in improving accuracy of the model. One should note, GA has a narrow exploration space in case of feature sub-spacing as compared to sub-sampling. Results suggest that ES is possibly useful in cases where maximum accuracy needs to be juiced out and computation is not an issue. It will also be interesting to see what happens if the algorithm is allowed to run for generations till the fitness test error reduces to zero. It's evident that better and more robust fitness functions need to be explored, even multi-objective fitness functions, which better represent generalizability and error of the ensemble. We plan to experiment with different predictors for each sampled datasets and explore how sub-spacing and sub-sampling can be combined into one algorithm.

Going with the theme of reproduction, we've released the basic framework for ES on GitHub (http://github.com/evoml/evoml). We encourage researchers to contribute to the project and test out different fitness functions themselves.

References

Breiman, Leo. Bagging predictors. *Machine learning*, 24 (2):123–140, 1996a.

Breiman, Leo. Bias, variance, and arcing classifiers. 1996b.

Breiman, Leo. Out-of-bag estimation. Technical report, Citeseer, 1996c.

Breiman, Leo. Pasting small votes for classification in large databases and on-line. *Machine Learning*, 36(1-2):85–103, 1999a.

Breiman, Leo. Using adaptive bagging to debias regressions. Technical report, Technical Report 547, Statistics Dept. UCB, 1999b.

Breiman, Leo. Random forests. *Machine learning*, 45(1): 5–32, 2001.

Büchlmann, Peter and Yu, Bin. Analyzing bagging. *Annals of Statistics*, pp. 927–961, 2002.

Domingos, Pedro M. Why does bagging work? a bayesian account and its implications. In *KDD*, pp. 155–158. Citeseer, 1997.

Fortin, Félix-Antoine, De Rainville, François-Michel, Gardner, Marc-André, Parizeau, Marc, and Gagné, Christian. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.

Freund, Yoav, Schapire, Robert E, et al. Experiments with a new boosting algorithm. In *ICML*, volume 96, pp. 148–156, 1996.

Gagné, Christian, Sebag, Michele, Schoenauer, Marc, and Tomassini, Marco. Ensemble learning for free with evolutionary algorithms? In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pp. 1782–1789. ACM, 2007.

Ho, Tin Kam. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):832–844, 1998.

Kuncheva, Ludmila I and Whitaker, Christopher J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.

Papakonstantinou, Periklis A, Xu, Jia, and Cao, Zhu. Bagging by design (on the suboptimality of bagging). In *AAAI*, pp. 2041–2047, 2014.

Tang, E Ke, Suganthan, Ponnuthurai N, and Yao, Xin. An analysis of diversity measures. *Machine Learning*, 65 (1):247–271, 2006.