Model clustering and resampling

Bart Bakker and Tom Heskes

Foundation for Neural Networks
Geert Grooteplein 21
6525 EZ Nijmegen
The Netherlands

Resampling is a popular approach to try and obtain better generalization performance with non-linear models. Individual models are trained on slightly different samples of the available data set, which are generated by e.g./ bootstrapping or cross-validation. The generalization performance obtained by the "average model" is usually better than the one that would result from training a single model on the full data set. "Bagging", based on data sets generated by bootstrapping and equal averaging of the individual models, is a typical example of such a resampling approach.

Despite many attempts, it is still not completely clear why and how bagging (and other, more advanced approaches) work so well. The general intuition is that the variation between the models induced by the resampling process is beneficial. This is by no means trivial. For example, for simple linear regression, resampling has no effect: the model obtained by averaging over many bootstrapped samples is equivalent to the single model trained on the full data set.

The key word is instability [1]. Resampling only helps in conjunction with unstable algorithms that yield "globally" rather than "locally" distinct solutions under small variations in initial conditions and data samples. Our working hypothesis is therefore that we need these "globally" distinct solutions to make bagging and other resampling approaches work.

To test this hypothesis, we have developed an algorithm for model clustering [2]. The goal is to summarize the ensemble of models *clusters*, each described by a *representative model*, or cluster center. Our clustering method is an extension of the approach introduced by Rose et al. [3] for unsupervised clustering. Clustering in this framework boils down to the minimization of a free energy. The main issue is how to measure the distance between the representative cluster center and the models in the ensemble. The most "natural" distance seems to be related to the training error. Here the cluster center plays the role of the model trained on the outcomes of the models in the ensemble. Clustering can then be implemented as a generalized EM algorithm. In the E-step, we compute the probabilities that link the models to the cluster centers. In the generalized

M-step the parameters of the representative cluster centers are fitted to the outcomes of the model in the ensemble, weighted with the probabilities computed in the E-step.

Our conjecture is that the cluster centers are necessary and sufficient to obtain improved generalization performance. Necessary, in the sense that no significant performance improvement is obtained with a single cluster. Sufficient, in the sense that we can replace the average over the resampled models by a (weighted) average over the cluster centers and obtain the same performance improvement.

Preliminary test results seem to validate our conjecture, especially the claim about sufficiency: there is no need to keep all the models in the ensemble. The cluster centers provide a sufficiently detailed summary. They have the obvious advantage of requiring less storage, but are also easier to analyze and interpret. For example, algorithms for determining the relevance of inputs can be applied on the cluster centers instead of on all individual models.

In some cases we find that a single "average model" quite accurately represents the ensemble average, and thus leads to about the same performance. This then seems to suggest that model averaging hardly helps (the models in the ensemble are only "locally" distinct, not "globally") and to validate the claim about necessity. An alternative interpretation is that the average model uses much more degrees of freedom than the individual models. For example, if the models are multi-layered perceptrons with n hidden units, the individual models effectively use only part of them (e.g., as a result of early stopping), whereas the average model can take advantage of the full model complexity.

Model clustering can also be used to quantize MCMC samples that result from Bayesian modeling. Furthermore, the models in the ensemble need not be the same. For example, we can train neural networks with different architectures and check whether networks with the same architecture end up in the same cluster.

References

[1] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[2] B. Bakker and T. Heskes. Model clustering by deterministic annealing. In M. Verleysen, editor, *Proceedings of the European Symposium on Artificial Neural Networks '99*, pages 87–92, 27 rue du Laekenveld - B 1080 Brussels - Belgium, 1999. D-Facto.

[3] K. Rose, E. Gurewitz, and G. Fox. Statistical mechanics of phase transitions in clustering. *Physical Review Letters*, 65:945–948, 1990.