3.6 Model Optimization and Tuning

The search for the best tuning parameter values can be done in many ways but most fall into two main categories: those that predefine which values to evaluate and those that incrementally determine the values. In the first case, the most well known procedure is *grid search*. Here, a set of candidate tuning parameter values are specified and then evaluated. In some cases, the model will have more than one tuning parameter and, in this case, a candidate parameter combination is multidimensional. We recommend using resampling to evaluate each distinct parameter value combination to get good estimates of how well each candidate performs. Once the results are calculated, the "best" tuning parameter combination is chosen and the final model is fit to the entire training set with this value. The best combination can be determined in various ways but the most common approach is to pick the candidate with the empirically best results.

As an example, a simple K-nearest neighbors model requires the number of neighbors. For the OkCupid data, this model will be used to predict the profession of the profile. In this context, the predictors contain many different types of profile characteristics, so that a "nearest neighbor" is really a similar profile based on many characteristics.

We will predefine the candidate set to be $K = 1, 3, \ldots 201$[28]. When combined with the same 10-fold cross-validation process, a total of 380 temporary models will be used to determine a good value of K. Once the best value of K is chosen, one final model is created using the optimal number of neighbors. The resampling profile is shown in Figure 3.11. Each black point shown in this graph is the average of performance for ten different models estimated using a distinct 90% of the training set. The configuration with the largest area under the ROC curve used 181 neighbors with a corresponding AUC of 0.805. Figure 3.11 shows the individual resampling profiles for each of the ten resamples. The reduced amount of variation in these data is mostly due to the size of the training set[29].

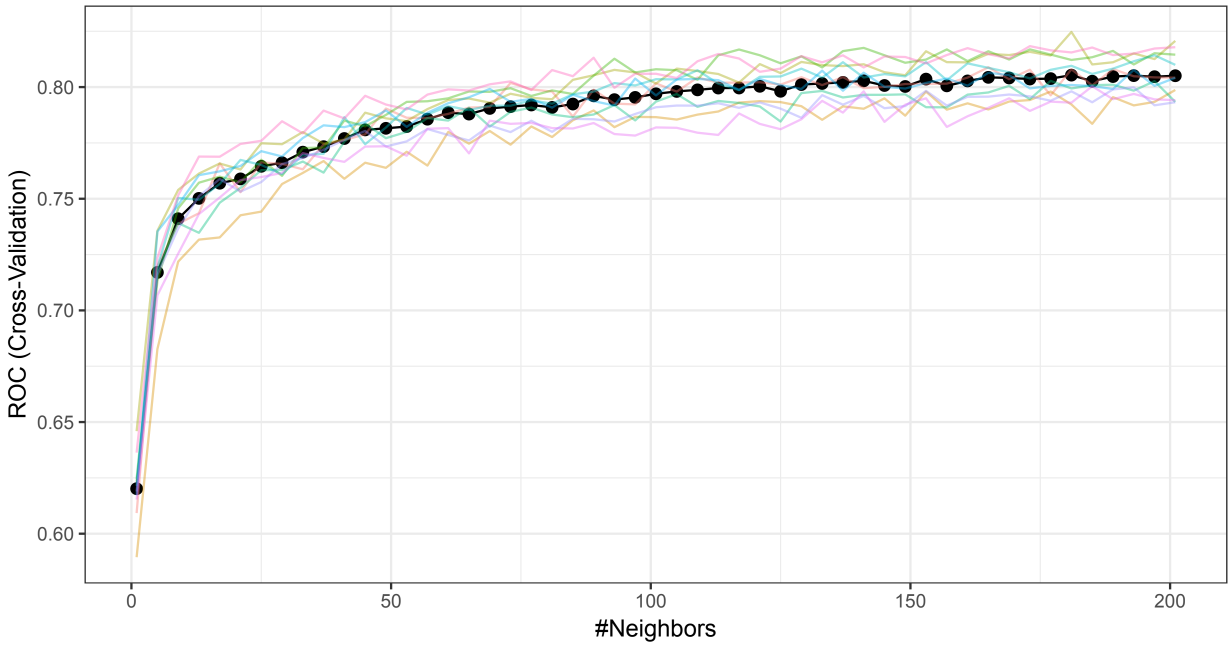

Figure 3.11: The resampling profile generated by a simple grid search on the number of neighbors in a K-NN classification model for the OkC data. The black line represents the averaged performance across 10 resamples while the other lines represent the performance across each individual resample.

When there are many tuning parameters associated with a model, there are several ways to proceed. First, a multidimensional grid search can be conducted where candidate parameter combinations and the grid of combinations are evaluated. In some cases, this can be very inefficient. Another approach is to define a range of possible values for each parameter and to randomly sample the multidimensional space enough times to cover a reasonable amount (Bergstra and Bengio 2012). This *random search* grid can then be resampled in the same way as a more traditional grid. This procedure can be very beneficial when there are a large number of tuning parameters and there is no *a priori* notion of which values should be used. A large grid may be inefficient to search, especially if the profile has a fairly stable pattern with little change over some range of the parameter. Neural networks, gradient boosting machines, and other models can effectively be tuned using this approach[30].

To illustrate this procedure, the OkCupid data was used once again. A single layer, feed-forward neural network was used to model the probability of being in the STEM field using the same predictors as the previous two models. This model is an extremely flexible nonlinear classification system with many tuning parameters. See Goodfellow, Bengio, and Courville (2016) for an excellent primer on neural networks and deep learning models.

The main tuning parameters for the model are:

- The number of hidden units. This parameter controls the complexity of the neural network. Larger values enable higher performance but also increase the risk of overfitting. For these data, the number of units in the hidden layers were randomly selected to be between 2 and 20.
- The activation function. The nonlinear function set by this parameter links different parts of the network. Three different functions were used: traditional sigmoidal curves, `tanh` , or rectified linear unit functions (ReLU).
- The dropout rate. This is the rate at which coefficients are randomly set to zero during training iterations and can attenuate overfitting (Nitish et al. 2014). Rates between 0 and 80% were considered.

The fitting procedure for neural network coefficients can be very numerically challenging. There are usually a large number of coefficients to estimate and there is a significant risk of finding a local optima. Here, we use a gradient-based optimization method called RMSProp[31] to fit the model. This is a modern algorithm for finding coefficient values and there are several model tuning parameters for this procedure[32]:

- The batch size controls how many of the training set data points are randomly exposed to the optimization process at each epoch (i.e. optimization iteration). This has the effect of reducing potential overfitting by providing some randomness to the optimization process. Batch sizes between 10 to 40K were considered.
- The learning rate parameter controls the rate of descent during the parameter estimation iterations and these values were contrasted to be between zero and one.
- A decay rate that decreases the learning rate over time (ranging between zero and one).
- The root mean square gradient scaling factor (ρ) controls how much the gradient is normalized by recent values of the squared gradient values. Smaller values of this parameter give more emphasis to recent gradients. The range of this parameter was set to be [0.0, 1.0].

For this model, 20 different seven dimensional tuning parameter combinations were created randomly using uniform distributions to sample within the ranges above. Each of these settings were evaluated using the same 10-fold cross-validation splits used previously.

The resampled ROC values significantly varied between the candidate parameter values. The best setting is italicized in Table 3.3 which had a corresponding area under the ROC curve of 0.838.

Table 3.3: The settings and results for a random search of the neural network parameter space.

Units	Dropout	Batch Size	Learn Rate	Grad. Scaling	Decay	Act. Fun.	ROC
7	*0.3368*	*11348*	*0.00385*	*0.55811*	*1.16e-04*	*sigmoid*	*0.838*
5	0.4023	36070	0.04142	0.95232	3.84e-02	sigmoid	0.837
12	0.6619	38206	0.25837	0.63325	3.09e-02	sigmoid	0.837
6	0.4720	38651	0.02634	0.80618	4.94e-05	sigmoid	0.834
7	0.3918	17235	0.01681	0.36270	2.90e-04	tanh	0.830
3	0.1190	16369	0.22210	0.22683	4.02e-02	relu	0.830
10	0.4979	19103	0.04818	0.83560	1.92e-03	relu	0.828
7	0.6139	38198	0.30864	0.68575	1.67e-03	sigmoid	0.824
2	0.3797	22255	0.10597	0.93841	4.27e-05	sigmoid	0.824
4	0.0694	18167	0.45844	0.94679	2.97e-03	tanh	0.811
14	0.6279	33800	0.18082	0.33286	1.08e-05	tanh	0.810
10	0.1466	33139	0.44443	0.72107	6.22e-05	tanh	0.810
20	0.0497	29465	0.40072	0.49598	7.07e-02	sigmoid	0.804
17	0.1068	17953	0.43256	0.87800	1.99e-04	tanh	0.794
13	0.5570	13558	0.13159	0.20389	5.96e-05	relu	0.793
7	0.6183	30606	0.82481	0.71944	2.61e-03	sigmoid	0.792
14	0.3866	30514	0.92724	0.38651	3.36e-03	tanh	0.782
2	0.0903	33439	0.63991	0.00398	1.92e-03	tanh	0.776
11	0.5909	32417	0.61446	0.63142	3.08e-04	tanh	0.731
10	0.4234	34655	0.49455	0.25216	3.05e-04	relu	0.698

As previously mentioned, grid search and random search methods have the tuning parameters specified in advance and the search does not adapt to look for novel values. There are other approaches that can be taken which do. For example, there are many nonlinear search methods such as the Nelder-Mead simplex search procedure, simulated annealing, and genetic algorithms that can be employed (Chong and Zak 2008). These methods conduct very thorough searches of the grid space but tend to be computationally expensive. One reason for this is that each evaluation of a new parameter requires a good estimate of performance to guide the search. Resampling is one of the best methods for doing this. Another approach to searching the space is called Bayesian optimization (Mockus 1994). Here, an initial pool of samples are evaluated using grid or random search. The optimization procedure creates a separate model to predict performance as a function of the tuning parameters and can then make a recommendation as to the next candidate set to evaluate. Once this new point is assessed, the model is updated and the process continues for a set number of iterations (Jones, Schonlau, and Welch 1998)[33].

One final point about the interpretation of resampling results is that, by choosing the best settings based on the results and representing the model's performance using these values, there is the risk of *optimization bias*. Depending on the problem, this bias might overestimate the model's true performance. There are nested resampling procedures that can be used to mitigate these biases. See Boulesteix and Strobl (2009) for more information.

References

Bergstra, J, and Y Bengio. 2012. "Random Search for Hyper-Parameter Optimization." *Journal of Machine Learning Research* 13:281–305.

Goodfellow, I, Y Bengio, and A Courville. 2016. *Deep Learning*. MIT Press.

Nitish, S, H Geoffrey, K Alex, S Ilya, and S Ruslan. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15:1929–58.

Chong, E, and S Zak. 2008. "Global Search Algorithms." In *An Introduction to Optimization*. John Wiley & Sons, Inc.

Mockus, J. 1994. “Application of Bayesian Approach to Numerical Methods of Global and Stochastic Optimization.” *Journal of Global Optimization* 4 (4). Springer:347–65.

Jones, D, M Schonlau, and W Welch. 1998. “Efficient Global Optimization of Expensive Black-Box Functions.” *Journal of Global Optimization* 13 (4). Springer:455–92.

Boulesteix, AL, and C Strobl. 2009. “Optimal Classifier Selection and Negative Bias in Error Rate Estimation: An Empirical Study on High-Dimensional Prediction.” *BMC Medical Research Methodology* 9 (1):85.

28. This number of neighbors is abnormally large compared to other data sets. This model tends to prefer values of K in, at most, the dozens.↩

29. This is generally not the case in other data sets; visualizing the individual profiles can often show an excessive amount of noise from resample-to-resample that is averaged out in the end.↩

30. However, a regular grid may be more efficient. For some models, there are optimizations that can be used to compute the results for a candidate parameter set that can be determined *without refitting* that models. The nature of random search cancels this benefit.↩

31. RMSprop is an general optimization method that uses gradients. The details are beyond the scope of this book but more information can be found in Goodfellow, Bengio, and

Courville (2016) and at `https://en.wikipedia.org/wiki/Stochastic_gradient_descent#RMSProp` .↩

32. Many of these parameters are fairly arcane for those not well acquainted with modern derivative-based optimization. Chapter 8 of Goodfellow, Bengio, and Courville (2016) has a substantial discussion of these methods.↩

33. The GitHub repository `http://bit.ly/2yjzB5V` has example code for nonlinear search methods and Bayesian optimization of tuning parameters.↩