sebastianraschka

ABOUT BOOKS ELSEWHERE PROJECTS RESOURCES BLOG

MACHINE LEARNING FAQ

Search

Machine Learning FAQ

Index

Which machine learning algorithms can be considered as among the best?

I recommend taking a look at

Wolpert, D.H., Macready, W.G. (1997), “No Free Lunch Theorems for Optimization”, IEEE Transactions on Evolutionary Computation 1, 67.

Unfortunately, there’s no real answer to this question: different datasets, questions, and assumptions require different algorithms – or in other words: we haven’t found the Master Algorithm, yet.

But let me write down thoughts about different classifiers at least:

- both logistic regression and SVMs work great for linear problems, logistic regression may be preferable for very noisy data
- naive Bayes may work better than logistic regression for small training set sizes; the former is also pretty fast, e.g., if you have a large multi-class problem, you’d only have to train one classifier whereas you’d have to use One-vs-Rest or One-vs-One with in SVMs or logistic regression (alternatively, you could

implement multinomial/softmax regression though); another point is that you don’t have to worry so much about hyperparameter optimization – if you are estimating the class priors from the training set, there are actually no hyperparameters

- kernel SVM/logistic regression is preferable for nonlinear data vs. the linear models
- k-nearest neighbor can also work quite well in practice for datasets with large number of samples and relatively low dimensionality
- Random Forests & Extremely Randomized trees are very robust and work well across a whole range of problems – linear and/or nonlinear problems

Personally, I tend to prefer a multi-layer neural network in most cases, given that my dataset is sufficiently large. In my experience, the generalization performance was almost always superior to one of the other approaches I listed above. But again, it really depends on the given dataset.