ISML-II: Machine Learning **Spring 2014**

Lecture 5- Bias Variance Trade-Off

Lecturer: Lorenzo Rosasco *Scribe: Lorenzo Rosasco*

Here we ask the question of how to choose K: is there an optima choice of K? Can it be computed in practice? Towards answering these questions we investigate theoretically the question of how K affects the performance of the KNN algorithm.

5.1 Tuning and Bias Variance Variance Decomposition

Ideally we would like to choose K that minimizes the expected error

$$\mathbf{E}_S\mathbf{E}_{x,y}(y - \hat{f}_K(x))^2.$$

We next characterize the corresponding minimization problem to uncover one of the most fundamental aspect of machine learning.
For the sake of simplicity we consider a regression model

$$y_i = f_*(x_i) + \delta_i, \quad \mathbf{E}\delta_I = 0, \mathbf{E}\delta_i^2 = \sigma^2 \quad i = 1, \dots, n$$

Moreover, we consider the least square loss function to measure errors, so that the performance of the KNN algorithm is given by the expected loss

$$\mathbf{E}_S\mathbf{E}_{x,y}(y - \hat{f}_K(x))^2 = \mathbf{E}_x \underbrace{\mathbf{E}_S\mathbf{E}_{y|x}(y - \hat{f}_K(x))^2}_{\varepsilon(K)}.$$

To get an insight on how to choose K, we analyze theoretically how this choice influences the expected loss. In fact, in the following we simplify the analysis considering the performance of KNN $\varepsilon(K)$ at a given point x.

First, note that

$$\varepsilon(K) = \sigma^2 + \mathbf{E}_S\mathbf{E}_{y|x}(f_*(x) - \hat{f}_K(x))^2,$$

where σ^2 can be seen as an irreducible error term. Second, to study the latter term we introduce the *expected* KNN algorithm,

$$\mathbf{E}_{y|x}\hat{f}_K(x) = \frac{1}{K}\sum_{\ell \in K_x} f_*(x_\ell).$$

We have

$$\mathbf{E}_S\mathbf{E}_{y|x}(f_*(x) - \hat{f}_K(x))^2 = \underbrace{(f_*(x) - \mathbf{E}_S\mathbf{E}_{y|x}\hat{f}_K(x))^2}_{Bias} + \underbrace{\mathbf{E}_S\mathbf{E}_{y|x}(\mathbf{E}_{y|x}\hat{f}_K(x) - \hat{f}_K(x))^2}_{Variance}$$

Finally, we have

$$\varepsilon(K) = \sigma^2 + (f_*(x) + \frac{1}{K}\sum_{\ell \in K_x} f_*(x_\ell))^2 + \frac{\sigma^2}{K}$$

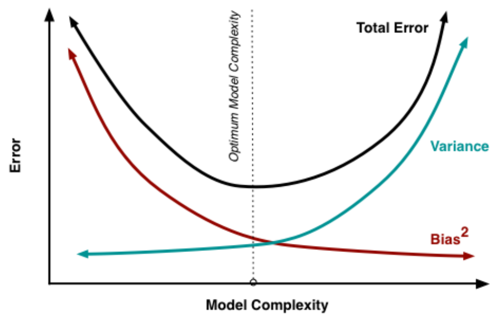

Figure 5.1. The Bias-Variance Tradeoff. In the KNN algorithm the parameter K controls the achieved (model) complexity.

5.2 The Bias Variance Trade-Off

We are ready to discuss the behavior of the (point-wise) expected loss of the KNN algorithm as a function of K. As it is clear from the above equation, the variance decreases with K. The bias is likely to increase with K, if the function f_* is suitably smooth. Indeed, for small K the few closest neighbors to x will have values close to $f_*(x)$, so their average will be close to $f_*(x)$. Whereas, as K increases neighbors will be further away and their average might move away from $f_*(x)$. A larger bias variance is preferred when data are few/noisy to achieve a better control of the variance, whereas the bias can be decreased as more data become available, hence reducing the variance. For any given training set, the best choice for K would be the one striking the optimal trade-off between bias and variance (that is the value minimizing their sum).

5.3 Cross Validation

While instructive, the above analysis is not directly useful in practice since the data distribution, hence the expected loss, is not accessible. In practice, data driven procedures are used to find a proxy for the expected loss. The simplest such procedure is called hold-out cross validation. Part of the training S set is hold-out, to compute a (holdout) error to be used as a proxy of the expected error. An empirical bias variance trade-off is achieved choosing the value of K that achieves minimum hold-out error. When data are scarce the hold-out procedure, based on a simple "two ways split" of the training set, might be unstable. In this case, so called V-fold cross validation is preferred, which is based on multiple data splitting. More precisely, the data are divided in V (non overlapping) sets. Each set is hold-out and used to compute an hold-out error which is eventually averaged to obtained the final V-fold cross validation error. The extreme case where $V = n$ is called leave-one-out cross validation.

5.3.1 Conclusions: Beyond KNN

Most of the above reasonings hold for a large class of learning algorithms beyond KNN. Indeed, many (most) algorithms depend one one or more parameter controlling the bias-variance tradeoff.