E0 370 Statistical Learning Theory **Lecture 16 (Oct 25, 2011)**

Consistency of Nearest Neighbor Methods

Lecturer: Shivani Agarwal *Scribe: Arun Rajkumar*

1 Introduction

In this lecture we return to the study of consistency properties of learning algorithms, where we will be interested in the question of whether the generalization error of the function learned by an algorithm approaches the Bayes error in the limit of infinite data. In particular, we will consider consistency properties of the simple k-nearest neighbor (k-NN) classification algorithm (in the next lecture, we will investigate consistency properties of algorithms such as SVMs and AdaBoost that effectively minimize a convex upper bound on the zero-one loss). We start with a brief review of statistical consistency and related results.

Recall from Lecture 10 that for any $h : \mathcal{X} \to \mathcal{Y}$, $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$, and probability distribution D on $\mathcal{X} \times \mathcal{Y}$, the *excess error* of h (w.r.t. D and ℓ) can be described as

$$\underbrace{\left(\mathrm{er}_D[h] - \mathrm{er}_D^{\ell,*}\right)}_{\text{excess error}} = \underbrace{\left(\mathrm{er}_D^{\ell}[h] - \mathrm{er}_D^{\ell}[\mathcal{H}]\right)}_{\text{estimation error}} + \underbrace{\left(\mathrm{er}_D^{\ell}[\mathcal{H}] - \mathrm{er}_D^{\ell,*}\right)}_{\text{approximation error}}; \tag{1}$$

here $\mathrm{er}_D^{\ell,*} = \inf_{h:\mathcal{X}\to\mathcal{Y}} \mathrm{er}_D^{\ell}[h]$ and $\mathrm{er}_D^{\ell}[\mathcal{H}] = \inf_{h\in\mathcal{H}} \mathrm{er}_D^{\ell}[h]$.

An algorithm $\mathcal{A} : \cup_{m=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^m \to \mathcal{H}$ (which given a training sample $S \in \cup_{m=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^m$ returns a function $h_S = \mathcal{A}(S) \in \mathcal{H}$) is *statistically consistent* in $\mathcal{H}$ w.r.t. D and ℓ if for all $\epsilon > 0$,

$$\mathbf{P}_{S\sim D^m}\left(\mathrm{er}_D^{\ell}[h_S] - \mathrm{er}_D^{\ell}[\mathcal{H}] \geq \epsilon\right) \longrightarrow 0 \;\text{ as } m\to\infty\,.$$

$\mathcal{A}$ is *universally consistent* in $\mathcal{H}$ w.r.t. ℓ if it is consistent in $\mathcal{H}$ w.r.t. D, ℓ for *all* distributions D.

Similarly, an algorithm $\mathcal{A} : \cup_{m=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^m \to \mathcal{Y}^{\mathcal{X}}$ is *Bayes consistent* w.r.t. D and ℓ if for all $\epsilon > 0$,

$$\mathbf{P}_{S\sim D^m}\left(\mathrm{er}_D^{\ell}[h_S] - \mathrm{er}_D^{\ell,*} \geq \epsilon\right) \longrightarrow 0 \;\text{ as } m\to\infty\,.$$

$\mathcal{A}$ is *universally Bayes consistent* w.r.t. ℓ if it is Bayes consistent w.r.t. D, ℓ for *all* distributions D.

We showed in Lecture 10 that:

(1) If $\mathcal{Y} = \{\pm 1\}$, $\ell = \ell_{0\text{-}1}$, and $\mathrm{VCdim}(\mathcal{H}) < \infty$, then empirical risk minimization (ERM) in $\mathcal{H}$ is universally consistent in $\mathcal{H}$ (in fact, one gets a fixed sample size/rate of convergence for all distributions D; this is what allowed us to establish that $\mathrm{VCdim}(\mathcal{H})$ finite $\implies$ $\mathcal{H}$ learnable by ERM in $\mathcal{H}$).

(2) If $\mathcal{Y} = \{\pm 1\}$, $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \ldots$, and $\mathrm{VCdim}(\mathcal{H}_i) < \mathrm{VCdim}(\mathcal{H}_{i+1})$ $\forall i$, then structural risk minimization (SRM) over $(H_i)_{i=1}^{\infty}$ is universally consistent in $\mathcal{H} = \cup_{i=1}^{\infty} \mathcal{H}_i$. This can yield universal consistency even for function classes $\mathcal{H}$ that do not have finite VC-dimension (but that can be decomposed as a hierarchy of finite VC-dimension classes as above).[1] In particular, if $\cup_{i=1}^{\infty} \mathcal{H}_i$ has zero approximation error, then SRM over $(H_i)_{i=1}^{\infty}$ is universally *Bayes* consistent.

While SRM can achieve Bayes consistency, it is typically not computationally feasible. Therefore, it is of interest to ask whether one can achieve Bayes consistency using other algorithms.

[1] Note however that here we do not get a fixed sample size/convergence rate that works for all D – in fact a fixed sample size in this case would contradict the result that $\mathcal{H}$ is learnable only if $\mathrm{VCdim}(\mathcal{H})$ is finite.

We will be interested in studying Bayes consistency primarily for binary classification algorithms. Therefore, for the rest of the lecture, we will fix $\mathcal{Y} = \{\pm 1\}$ and $\ell = \ell_{0\text{-}1}$, and will let er_D^* abbreviate $\mathrm{er}_D^{0\text{-}1,*}$. In this setting, define the conditional label probability $\eta : \mathcal{X} \to [0,1]$ as

$$\eta(x) = \mathbf{P}(y = 1|x)\,. \tag{2}$$

Note that η depends on the distribution D. Now define $h^* : \mathcal{X} \to \{\pm 1\}$ as

$$h^*(x) = \begin{cases} 1 & \text{if } \eta(x) \geq \frac{1}{2} \\ -1 & \text{otherwise.} \end{cases} \tag{3}$$

Then clearly[2]

$$\mathrm{er}_D[h^*] = \mathrm{er}_D^*\,. \tag{4}$$

Any such classifier that achieves the Bayes error is called a *Bayes classifier*.

In the following, we will consider the k-nearest neighbor algorithm and ask whether the error of the classifier returned by k-NN approaches the Bayes error er_D^*.

2 k-Nearest Neighbor Algorithm

The k-nearest neighbor algorithm is one of the simplest machine learning algorithms for classification. The basic idea of the algorithm is the following: to predict the class label of a new instance, take a majority vote among the classes of the k points in the training sample that are closest to this instance. Of course, 'closest' requires a notion of distance among instances. The most common setting is one where the instances are vectors in $\mathbb{R}^n$, and the distance used is the Euclidean distance.

Formally, let $\mathcal{X} \subseteq \mathbb{R}^n$. The k-NN algorithm receives as input $S = ((\mathbf{x}_1, y_1), \ldots (\mathbf{x}_m, y_m)) \in (\mathcal{X} \times \{\pm 1\})^m$ and produces a classification rule $h_S : \mathcal{X} \to \{\pm 1\}$ given by the following:

$$h_S(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^m y_i w_i(\mathbf{x}) \geq 0 \\ -1 & \text{otherwise} \end{cases} \;=\; \begin{cases} 1 & \text{if } \sum_{i:y_i=1} w_i(\mathbf{x}) \geq \sum_{i:y_i=-1} w_i(\mathbf{x}) \\ -1 & \text{otherwise} \end{cases} \tag{5}$$

where

$$w_i(\mathbf{x}) = \begin{cases} \frac{1}{k} & \text{if } \mathbf{x}_i \text{ is one of the } k \text{ nearest neighbors of } \mathbf{x} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

In determining nearest neighbors, any fixed method can be used to break ties (e.g. a common method is to take the point with smaller index as the closer neighbor: if two training points $\mathbf{x}_i, \mathbf{x}_j$ are equidistant from the point $\mathbf{x}$ to be classified, $\|\mathbf{x}_i - \mathbf{x}\| = \|\mathbf{x}_j - \mathbf{x}\|$, then $\mathbf{x}_i$ is treated as being closer to $\mathbf{x}$ than $\mathbf{x}_j$ if $i < j$, and vice-versa). Note that the weights $w_i(\mathbf{x})$ defined above depend on all the training points $\mathbf{x}_1, \ldots \mathbf{x}_m$.

The question of interest to us is: how does the k-NN algorithm behave in the limit of infinite data? We start with some classical results on convergence properties of the k-NN algorithm when k is fixed.

Consider first the special case $k = 1$. Let

$$\mathrm{er}_D^{\mathrm{NN}} = 2\mathbf{E}_{\mathbf{x}}[\eta(\mathbf{x})(1 - \eta(\mathbf{x}))]\,. \tag{7}$$

Then we have the following classical result of Cover and Hart [1]:

Theorem 2.1 (Cover and Hart, 1967)**.** Let D be any probability distribution on $\mathcal{X} \times \{\pm 1\}$. Then the 1-NN algorithm satisfies

$$\mathbf{E}_{S \sim D^m}\left[\mathrm{er}_D[h_S]\right] \to \mathrm{er}_D^{\mathrm{NN}} \quad \text{as } m \to \infty.$$

Moreover, the following relations hold:

$$\mathrm{er}_D^* \leq \mathrm{er}_D^{\mathrm{NN}} \leq 2\mathrm{er}_D^*(1 - \mathrm{er}_D^*) \leq 2\mathrm{er}_D^*\,.$$

[2]Exercise: verify that $\mathrm{er}_D[h^*] \leq \mathrm{er}_D[h]\ \forall h : \mathcal{X} \to \mathcal{Y}$.

It follows from the above result that if $\mathrm{er}_D^* = 0$ (i.e. the labels y are a deterministic function of the instances $\mathbf{x}$), then the 1-NN algorithm is Bayes consistent w.r.t. D (and $\ell_{0\text{-}1}$). In general, however, $\mathrm{er}_D^{\mathrm{NN}}$ may be different from er_D^*, and 1-NN need not be Bayes consistent.

A similar result holds for any fixed integer $k > 1$. In this case, let

$$\mathrm{er}_D^{k\mathrm{NN}} = \begin{cases} \mathbf{E}_{\mathbf{x}}\left[\sum_{j=0}^{k}\eta(\mathbf{x})^j(1-\eta(\mathbf{x}))^{k-j}\left(\eta(\mathbf{x})\cdot\mathbf{1}\Big(j<\frac{k}{2}\Big)+(1-\eta(x))\cdot\mathbf{1}\Big(j>\frac{k}{2}\Big)\right)\right] & \text{if } k \text{ is odd} \\ \mathrm{er}_D^{(k-1)\mathrm{NN}} & \text{if } k \text{ is even.} \end{cases} \tag{8}$$

Then we have:

Theorem 2.2. Let D be any probability distribution on $\mathcal{X} \times \{\pm 1\}$. Let $k > 1$. Then the k-NN algorithm satisfies

$$\mathbf{E}_{S\sim D^m}\Big[\mathrm{er}_D[h_S]\Big] \to \mathrm{er}_D^{k\mathrm{NN}} \text{ as } m\to\infty.$$

Moreover, the following relations hold:

$$\mathrm{er}_D^* \leq \ldots \leq \mathrm{er}_D^{5\mathrm{NN}} \leq \mathrm{er}_D^{3\mathrm{NN}} \leq \mathrm{er}_D^{1\mathrm{NN}} \leq 2\mathrm{er}_D^* .$$

Again, if $\mathrm{er}_D^* = 0$, then the k-NN algorithm with fixed $k > 1$ is Bayes consistent w.r.t. D, but in general, even with a large (but fixed) value for k, the algorithm need not be Bayes consistent.

Below we will see that by allowing the choice of k to vary with the number of training examples m (in particular, by allowing k to increase slowly with m), the nearest neighbor algorithm can actually be made a universally Bayes consistent algorithm. This follows from the celebrated theorem of Stone [3], which was the first result to establish universal Bayes consistency for any learning algorithm for binary classification.

3 Stone's Theorem

We will describe Stone's theorem for a more general class of weighted-average plug-in classification methods, of which the nearest neighbor algorithm can be viewed as a special case.

3.1 Plug-in Classifiers

Recall the form of the Bayes classifier h^* in Eq. (3). A natural approach to learning a classifier from a sample S is then to approximate the conditional probabilities $\eta(x)$ via an estimate $\eta_S : \mathcal{X}\to[0,1]$, and to 'plug in' this estimate in Eq. (3) to get a classifier of the form

$$h_S(x) = \begin{cases} 1 & \text{if } \eta_S(x) \geq \frac{1}{2} \\ -1 & \text{otherwise.} \end{cases} \tag{9}$$

Intuitively, one would expect that if η_S is a good approximation to η, then the plug-in classifier h_S above should also be a good approximation h^*, i.e. should have generalization error close to the Bayes error. This is captured in the following result:

Theorem 3.1. Let D be any probability distribution on $\mathcal{X} \times \{\pm 1\}$, η be the corresponding conditional probability function, $\eta_S : \mathcal{X}\to[0,1]$ be any estimate of η, and h_S be defined as in Eq. (9). Then

$$\mathrm{er}_D[h_S] - \mathrm{er}_D^* \;\leq\; 2\mathbf{E}_x\Big[|\eta_S(x) - \eta(x)|\Big] .$$

Proof. We have,

$$\begin{aligned}
\mathrm{er}_D[h_S] - \mathrm{er}_{D^*} &= \mathbf{E}_{(x,y)}\Big[\mathbf{1}(h_S(x) \neq y) - \mathbf{1}(h^*(x) \neq y)\Big] && (10)\\
&= \mathbf{E}_x\Big[\mathbf{E}_{y|x}\Big[\mathbf{1}(y=1)\Big(\mathbf{1}(h_S(x)=-1) - \mathbf{1}(h^*(x)=-1)\Big) \\
&\qquad +\mathbf{1}(y=-1)\Big(\mathbf{1}(h_S(x)=1) - \mathbf{1}(h^*(x)=1)\Big)\Big]\Big] && (11)\\
&= \mathbf{E}_x\Big[\eta(x)\Big(\mathbf{1}(h_S(x)=-1) - \mathbf{1}(h^*(x)=-1)\Big) \\
&\qquad +(1-\eta(x))\Big(\mathbf{1}(h_S(x)=1) - \mathbf{1}(h^*(x)=1)\Big)\Big] && (12)\\
&= \mathbf{E}_x\Big[(2\eta(x)-1)\Big(\mathbf{1}(h_S(x)=-1) - \mathbf{1}(h^*(x)=-1)\Big)\Big] && (13)\\
&= \mathbf{E}_x\Big[(2\eta(x)-1)\Big(\mathbf{1}(h_S(x)=-1, h^*(x)=1) - \mathbf{1}(h_S(x)=1, h^*(x)=-1)\Big)\Big] && (14)\\
&= \mathbf{E}_x\Big[|2\eta(x)-1| \cdot \mathbf{1}(h_S(x) \neq h^*(x))\Big] && (15)\\
&\leq 2\,\mathbf{E}_x\Big[|\eta_S(x) - \eta(x)|\Big], && (16)
\end{aligned}$$

since $h_S(x) \neq h^*(x) \implies |\eta_S(x) - \eta(x)| \geq |\eta(x) - \frac{1}{2}|$. □

Corollary 3.2. Under the conditions of Theorem 3.1,

$$\mathrm{er}_D[h_S] - \mathrm{er}^*_D \;\leq\; 2\sqrt{\mathbf{E}_x\Big[(\eta_S(x) - \eta(x))^2\Big]}\,.$$

Proof. This follows directly from Theorem 3.1 by Jensen's inequality:

$$\Big(\mathbf{E}_x\Big[|\eta_S(x) - \eta(x)|\Big]\Big)^2 \;\leq\; \mathbf{E}_x\Big[(\eta_S(x) - \eta(x))^2\Big]\,.$$

□

The above results imply that in order to prove Bayes consistency of a plug-in classifier h_S, it suffices to prove convergence of the corresponding conditional probability estimate η_S; in particular, it suffices to show either $\mathbf{E}_{S\sim D^m}\Big[\mathbf{E}_x\Big[|\eta_S(x) - \eta(x)|\Big]\Big] \to 0$ or $\mathbf{E}_{S\sim D^m}\Big[\mathbf{E}_x\Big[(\eta_S(x) - \eta(x))^2\Big]\Big] \to 0$ as $m \to \infty$.

3.2 Weighted-Average Plug-in Classifiers

Consider now a specific family of plug-in classifiers termed 'weighted-average' plug-in classifiers, which estimate the conditional probability $\eta(x)$ via a weighted average of the training labels. Specifically, let $\mathcal{X} \subseteq \mathbb{R}^n$. Given a sample $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)) \in (\mathcal{X} \times \{\pm 1\})^m$ and an instance $\mathbf{x} \in \mathcal{X}$, let

$$\eta_S(\mathbf{x}) = \sum_{i=1}^{m} y'_i w_i(\mathbf{x})\,, \tag{17}$$

where $y'_i = (y_i + 1)/2 \in \{0, 1\}$, and the weights $w_i(\mathbf{x})$ can depend on $\mathbf{x}_1, \ldots, \mathbf{x}_m$ and must be non-negative and sum to one: $w_i(\mathbf{x}) \geq 0\ \forall i, \mathbf{x}$ and $\sum_{i=1}^m w_i(\mathbf{x}) = 1\ \forall \mathbf{x}$. Then the weighted-average plug-in classifier h_S with weights $w_i(\mathbf{x})$ is obtained by using this estimate η_S in Eq. (9).

Clearly, the k-NN algorithm can be viewed as a special case of the weighted-average plug-in classifier, with weights given by $w_i(\mathbf{x}) = 1/k$ if $\mathbf{x}_i$ is one of the k nearest neighbors of $\mathbf{x}$ in S, and $w_i(\mathbf{x}) = 0$ otherwise. Stone's theorem establishes universal Bayes consistency of such classifiers provided the weights satisfy certain conditions; we will see that if k is allowed to vary with m such that $k_m \to \infty$ and $\frac{k_m}{m} \to 0$ as $m \to \infty$, then the weights associated with the resulting k_m-NN classifier satisfy the required conditions.[3]

[3]The k_m-NN algorithm under these conditions on k_m had been previously known to be Bayes consistent under certain regularity conditions on the distribution D, but Stone's theorem was the first to establish universal Bayes consistency.

3.3 Consistency of Weighted-Average Plug-in Classifiers: Stone's Theorem

Theorem 3.3 (Stone, 1977)**.** Let $\mathcal{X} \subseteq \mathbb{R}^n$, and let h_S be a weighted-average plug-in classifier as above with weights w_i such that for all distributions μ on $\mathcal{X}$, the following conditions hold:

(i) $\exists\, c > 0$ such that for every non-negative measurable function $f : \mathcal{X} \to \mathbb{R}$ with $\mathbf{E}_{\mathbf{x}}[f(\mathbf{x})] < \infty$,

$$\mathbf{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_m,\mathbf{x}}\left[\sum_{i=1}^{m} w_i(\mathbf{x}) f(\mathbf{x})\right] \;\leq\; c\,\mathbf{E}_{\mathbf{x}}[f(\mathbf{x})]\,; \tag{18}$$

(ii) For all $a > 0$,

$$\mathbf{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_m,\mathbf{x}}\left[\sum_{i=1}^{m} w_i(\mathbf{x})\,\mathbf{1}\Big(\|\mathbf{x}_i - \mathbf{x}\| \geq a\Big)\right] \to 0 \text{ as } m\to\infty; \quad \text{and} \tag{19}$$

(iii)

$$\mathbf{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_m,\mathbf{x}}\left[\max_{1\leq i\leq m} w_i(\mathbf{x})\right] \to 0 \text{ as } m\to\infty. \tag{20}$$

Then (an algorithm which given S, outputs) h_S is universally Bayes consistent.

Proof. The proof proceeds by showing that

$$\mathbf{E}_{S\sim D^m}\Big[\mathbf{E}_{\mathbf{x}}\Big[(\eta_S(\mathbf{x}) - \eta(\mathbf{x}))^2\Big]\Big] \to 0 \text{ as } m\to\infty. \tag{21}$$

Towards this, let

$$\hat{\eta}_S(\mathbf{x}) = \sum_{i=1}^{m} w_i(\mathbf{x})\eta(\mathbf{x}_i)\,. \tag{22}$$

Then

$$\begin{aligned}
\mathbf{E}_{S,\mathbf{x}}\Big[(\eta_S(\mathbf{x}) - \eta(\mathbf{x}))^2\Big] &= \mathbf{E}_{S,\mathbf{x}}\Big[(\eta_S(\mathbf{x}) - \hat{\eta}_S(\mathbf{x}) + \hat{\eta}_S(\mathbf{x}) - \eta(\mathbf{x}))^2\Big] &(23)\\
&\leq 2\left(\mathbf{E}_{S,\mathbf{x}}\Big[(\eta_S(\mathbf{x}) - \hat{\eta}_S(\mathbf{x}))^2\Big] + \mathbf{E}_{S,\mathbf{x}}\Big[(\hat{\eta}_S(\mathbf{x}) - \eta(\mathbf{x}))^2\Big]\right), &(24)
\end{aligned}$$

since $(a+b)^2 \leq 2(a^2+b^2)$. Consider first the first term in the RHS of Eq. (24); we will show this converges to 0 as $m\to\infty$. We have,

$$\begin{aligned}
\mathbf{E}_{S,\mathbf{x}}\Big[(\eta_S(\mathbf{x}) - \hat{\eta}_S(\mathbf{x}))^2\Big] &= \mathbf{E}_{S,\mathbf{x}}\left[\left(\sum_{i=1}^{m} w_i(\mathbf{x})(y_i' - \eta(\mathbf{x}_i))\right)^2\right] &(25)\\
&= \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbf{E}_{S,\mathbf{x}}\Big[w_i(\mathbf{x})w_j(\mathbf{x})(y_i' - \eta(\mathbf{x}_i))(y_j' - \eta(\mathbf{x}_j))\Big] &(26)
\end{aligned}$$

Now for $i \neq j$,

$$\begin{aligned}
\mathbf{E}_{S,\mathbf{x}}\Big[w_i(\mathbf{x})w_j(\mathbf{x})(y_i' - \eta(\mathbf{x}_i))(y_j' - \eta(\mathbf{x}_j))\Big] &= \mathbf{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_m,\mathbf{x}}\Big[\mathbf{E}_{y_i,y_j|\mathbf{x}_1,\ldots,\mathbf{x}_m,\mathbf{x}}\Big[w_i(\mathbf{x})w_j(\mathbf{x})(y_i' - \eta(\mathbf{x}_i))(y_j' - \eta(\mathbf{x}_j))\Big]\Big]\\
&= 0\,. &(27)
\end{aligned}$$

Therefore we have

$$\begin{aligned}
\mathbf{E}_{S,\mathbf{x}}\Big[(\eta_S(\mathbf{x}) - \hat{\eta}_S(\mathbf{x}))^2\Big] &= \sum_{i=1}^{m} \mathbf{E}_{S,\mathbf{x}}\Big[w_i^2(\mathbf{x})(y'_i - \eta(\mathbf{x}_i))^2\Big] && (28)\\
&\leq \sum_{i=1}^{m} \mathbf{E}_{S,\mathbf{x}}\Big[w_i^2(\mathbf{x})\Big] \quad (\text{since } (y'_i - \eta(\mathbf{x}_i))^2 \leq 1) && (29)\\
&= \mathbf{E}_{S,\mathbf{x}}\left[\sum_{i=1}^{m} w_i^2(\mathbf{x})\right] && (30)\\
&\leq \mathbf{E}_{S,\mathbf{x}}\left[\left(\max_{1\leq i\leq m} w_i(\mathbf{x})\right)\left(\sum_{j=1}^{m} w_j(\mathbf{x})\right)\right] && (31)\\
&= \mathbf{E}_{S,\mathbf{x}}\left[\max_{1\leq i\leq m} w_i(\mathbf{x})\right] && (32)\\
&\longrightarrow 0 \text{ as } m\to\infty, \text{ by condition (iii).} && (33)
\end{aligned}$$

The second term in the RHS of Eq. (24) can also be shown to converge to zero; this makes use of conditions (i) and (ii) (see [2] for details). Together with the above, this yields the desired result. □

As noted above, it can be shown that with k_m chosen so that $k_m\to\infty$ and $\frac{k_m}{m}\to 0$ as $m\to\infty$, the k_m-NN classifier satisfies the conditions of Stone's theorem, thus establishing universal Bayes consistency for such classifiers. Details can be found in [2].

4 Additional Pointers

4.1 'Classification is Easier than Regression'

The following result holds for plug-in classifiers using a conditional probability estimate η_S satisfying $\mathbf{E}_{S,x}\big[(\eta_S(x) - \eta(x))^2\big]\to 0$ as $m\to\infty$:

Theorem 4.1. Let $\eta_S(x)$ be such that

$$\mathbf{E}_{S,x}\Big[(\eta_S(x) - \eta(x))^2\Big]\to 0 \text{ as } m\to\infty,$$

and let

$$h_S(x) = \begin{cases} 1 & \text{if } \eta_S(x) \geq \frac{1}{2} \\ -1 & \text{otherwise.} \end{cases}$$

Then

$$\frac{\mathrm{er}_D[h_S] - \mathrm{er}_D^*}{\sqrt{\mathbf{E}_{S,x}\Big[(\eta_S(x) - \eta(x))^2\Big]}} \longrightarrow 0 \text{ as } m\to\infty.$$

Note that the actual value of the above ratio cannot be bounded. Further details can be found in [2].

4.2 Slow Rates of Convergence

Theorem 4.2. Let $\{a_n\}$ be a sequence of positive numbers converging to 0, with $\frac{1}{16} \geq a_1 \geq a_2 \geq \ldots$. Then for any classification method, there exists a distribution D on $\mathcal{X}\times\{\pm 1\}$ with $\mathrm{er}_D^* = 0$ such that

$$\mathbf{E}_{S\sim D^m}\Big[\mathrm{er}_D[h_S]\Big] \geq a_m \quad \forall m\,.$$

This implies that there are no universal rates of convergence w.r.t. er_D^* for any method; any such rates must come with some restrictions on D. See [2] for further details.

5 Next Lecture

In the next lecture, we will look at some recent results on consistency of learning algorithms that minimize a convex surrogate of the zero-one loss, such as support vector machines and AdaBoost.

References

[1] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[2] Luc Devroye, Laszlo Gyorfi, and Gabor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[3] Charles J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–620, 1977.