Basic Statistical Learning - Part II

Arash Goligerdian and Yerbol Palzhanov

1 Concentration of Measure Inequalities

Concentration of measure is a phenomenon that is related to the classical law of large numbers.

1.1 McDiarmid's Bound

Roughly speaking, McDiarmid's bound states that if arbitrary replacements of random variables ξ_i do not affect the value of the random variable $g(\xi_1,\ldots,\xi_m)$ excessively, then g is concentrated.

Theorem 1 (McDiarmid). *Let $\xi_1,\ldots,\xi_m$ be i.i.d. random variables and assume that there exists a function $g : \xi^m \to \mathbb{R}$ with the property that for all $i \in [m]$ (we use the shorthand $[m] := 1,\ldots,m$), and $c_i > 0$,*

$$\sup_{\xi_1,\ldots\xi_m,\xi_i'\in\xi} |g(\xi_1,\ldots\xi_m) - g(\xi_1,\ldots\xi_{i-1},\xi_i',\xi_{i+1},\ldots,\xi_m)| \leq c_i \tag{1}$$

where ξ_i' is drawn from the same distribution as ξ_i. Then

$$P\{|g(\xi_1,\cdots\xi_m) - \mathbf{E}(g(\xi_1,\cdots\xi_m))| > \varepsilon\} \leq 2\exp\Big(-\frac{2\varepsilon^2}{\sum_{i=1}^m c_i^2}\Big). \tag{2}$$

This means that a bound similar to the law of the large numbers can be applied to any function g which does not overly depend on individual samples ξ_i. Returning to the example of the sample mean, if we define

$$g(\xi_1,\cdots\xi_m) := \frac{1}{m}\sum_{i=1}^m \xi_i$$

where $\xi \in [a,b]$, then, clearly, $c_i = \frac{1}{m}(b-a)$ since f can only change by this amount if one particular sample is replaced by another. This mean that the rhs of (2) becomes $2\exp\Big(-\frac{2m\varepsilon^2}{(b-a)^2}\Big)$, in other words, we recover Hoeffding's bound as a special case.

1.2 Uniform Stability and Convergence

In order to apply these bounds to learning algorithms we must introduce the notion of *uniform stability.* This is to determine the amount by which an estimate $f : X \to Y$ based on the training data $D := \{(x_1,y_1),\ldots,(x_m,y_m)\} \subset X \times Y$ changes if we change one of the training patterns.

Definition 1 (Uniform Stability). Denote a training sample of size m by D. Moreover, denote by $D^i := (D\backslash\{z_i\}) \cup \{z\}$, where $z = (x,y)$ is the training sample with the i^{th} observation replaced by z. Finally, denote by f_Z the estimate produced by our learning algorithm of choice (and likewise by f_{Z^i} the estimate based on Z_i). We call this mapping $Z \to f_Z$ uniformly β-stable with respect to a loss function L if

$$|L(x,y,f_Z(x)) - L(x,y,f_{Z^i}(x))| \leq \beta \text{ for all } (x,y) \in X \times Y, \text{ all } Z, \text{ and all } i. \tag{3}$$

This means that the loss due to the estimates generated from Z, where an arbitrary pattern of the sample has been replaced, will not differ anywhere by more than β.

As we shall see, the notion of uniform stability is satisfied for regularization networks of different types, provided that the loss function L is Lipschitz continuous. The following theorem uses Theorem 1 to prove that β-stable algorithms exhibit uniform convergence of the empirical risk $R_{emp}[f]$ to the expected risk $R[f]$.

Theorem 2 (Bousquet and Elisseeff). *Assume that we have β-stable algorithm with the additional requirement that $f_Z(x) \leq M$ for all $x \in X$ and for all training samples $Z \subset X \times Y$. Then, for $m \geq \frac{8M^2}{\varepsilon^2}$, we have*

$$P\{|R_{emp}[f_Z] - R[f_Z]| > \varepsilon\} \leq \frac{64Mm\beta + 8M^2}{m\varepsilon^2} \tag{4}$$

and for any $m \geq 1$

$$P\{|R_{emp}[f_Z] - R[f_Z]| > \varepsilon + \beta\} \leq 2\exp\Big(-\frac{m\varepsilon^2}{2(m\beta + M)^2}\Big). \tag{5}$$

This means that if β decreases with increasing m, or, in particular, if $\beta = O(m^{-1})$, then we obtain bounds that are optimal in their rate of convergence, specifically, bounds which have the same convergence rate as Hoeffding's bound .

To keep matters simple, we only the second inequality

Proof We first give a bound on the expected difference between $R_{emp}[f_Z]$ and $R[f_Z]$ (hence the bias term) and subsequently will bound the variance. This leads to

$$|\mathbf{E}_Z\big[R_{emp}[f_Z]-R[f_Z]\big]| = \Big|\mathbf{E}_{Z,z}\Big[\frac{1}{m}\sum_{i=1}^{m} L(x_i,y_i,f_Z(x_i)) - L(x,y,f_Z(x))\Big]\Big|$$
$$= \Big|\mathbf{E}_Z\Big[\frac{1}{m}\sum_{i=1}^{m} L(x,y,f_{Z^i}(x)) - L(x,y,f_Z(x))\Big]\Big| \leq \beta \qquad (6)$$

The last equality (8) followed from the fact that, since we are taking the expectation over Z,z, we may as well replace z_i by z in the terms stemming from the empirical error. The bound then follows from the assumption that we have a uniformly β-stable algorithm.

Now that we have a bound on the expectation, we deal with the variance. Since we want to apply Theorem 1, we have to analyze the deviations of $(R_{emp}[f_Z]-R[f_Z])$ from $(R_{emp}[f_{Z^i}]-R[f_{Z^i}])$.

$$\big|(R_{emp}[f_Z]-R[f_Z]) - (R_{emp}[f_{Z^i}]-R[f_{Z^i}])\big| \leq \qquad (7)$$
$$\big|R[f_Z]-R[f_{Z^i}]\big| + \big|R_{emp}[f_Z]-R_{emp}[f_{Z^i}]\big| \leq \qquad (8)$$
$$\beta + \frac{1}{m}|L(x_i,y_i,f_Z(x_i)) - L(x,y,f_{Z^i}(x))| +$$
$$+\frac{1}{m}\sum_{j\neq i}^{m}\big|L(x_j,y_j,f_Z(x_j)) - L(x_j,y_j,f_{Z^i}(x_j))\big| \leq \beta + \frac{2M}{m} + \beta \qquad (9)$$

Here the second inequality follows from the triangle inequality and the fact that the learning algorithm is β-stable. Finally, we split the empirical risks into their common parts depending on Z^i and the remainder. From the last inequality it follows that $L_i = 2\frac{\beta m + M}{m}$ as required by Theorem 1. This, in combination with (6), completes the proof. □

1.3 Uniform Stability of Regularization Networks

We next show that the learning algorithms we have been studying so far actually satisfy Definition 1 and compute the corresponding value of β.

Theorem 3 (Algorithmic Stability of Risk Minimizers). *The algorithm minimizing the regularized risk functional R_{reg}*

$$R_{reg}[f] := R_{emp}[f] + \frac{\lambda}{2}||f||^2 = \frac{1}{m}\sum_{i=1}^{m} L(x_i,y_i,f(x_i)) + \frac{\lambda}{2}||f||^2 \qquad (10)$$

has stability $\beta = \frac{2C^2k^2}{m\lambda}$, where k is a bound on $||k(x,\cdot)|| = \sqrt{k(x,x)}$, L is a convex loss function, $||\cdot||$ is the RKHS norm induced by k, and C is a bound on the Lipschitz constant of the loss function $L(x,y,f(x))$, viewed as a function of $f(x)$.

We can see that the stability β of the algorithm depends on the regularization constant via $\frac{1}{\lambda m}$ and into Theorem 4 below gives a useful convergence bound, hence we

may be able to afford to choose weaker regularization if the sample size increases. For many estimators, such as Support Vector Machines, we use a constant value of $C = \frac{1}{\lambda m}$. In the context of algorithmic stability this means that we effectively use algorithms with the same stability, regardless of the sample size.

2 Leave-One-Out Estimates

Rather than betting on the proximity between the empirical risk and the expected risk we may make further use of the training data and compute what is commonly referred to as the *leave-one-out error* of a sample. The basic idea is that we find an estimate f^i from a sample consisting of $m-1$ patterns by leaving the i^{th} pattern out and, subsequently, compute the error of mis-prediction on (x^i, y^i). The error is then averaged over all m possible patterns. The hope is that such a procedure will provide us with a quantity that is very closely related to the real expected error.

Before we delve into the practical details of estimating the leave-one-out error, we need a formal definition and have to prove that the leave-one-out estimator is a useful quantity.

Definition 2 (Leave-One-Out Error). Denote by f_Z the estimate obtained by a learning algorithm, given the sample Z, by $Z^i := Z' \backslash \{(x_i, y_i)\}$ the sample obtained by removing the i^{th} pattern, and by f_{Z^i} the corresponding estimate, obtained by the same learning algorithm (note that we changed the definition of Z^i from that in the previous section). Then the leave-one-out error is defined as

$$R_{LOO} := \frac{1}{m} \sum_{i=1}^{m} L(x_i, y_i, f_{Z^i}(x_i)). \tag{11}$$

The following theorem by Luntz and Brailovsky shows that $R_{LOO}(Z)$ is an almost unbiased estimator.

Theorem 4 (Leave-One-Out Error is Almost Unbiased). *Denote by P a distribution over $X \times Y$, and by Z_m and Z_{m-1} samples of size m and $m-1$ respectively, drawn i.i.d. from P. Moreover, denote by $R[f_{Z_{m-1}}]$ the expected risk of an estimator derived from the sample Z_{m-1}. Then, for any learning algorithm, the leave-one-out error is almost unbiased,*

$$\boldsymbol{E}_{Z_{m-1}}[R[f_{Z_{m-1}}]] = \boldsymbol{E}_{Z_m}[R_{LOO}(Z_m)]. \tag{12}$$

Proof We begin by rewriting $\mathbf{E}_{Z_{m-1}}[R[f_{Z_{m-1}}]]$ in terms of expected values only. By definition $R(f) := \mathbf{E}[L(x, y, f(x))]$ and therefore, the lhs of (14) can be written as

$$\mathbf{E}_{Z_{m-1}}[R[f_{Z_{m-1}}]] = \mathbf{E}_{Z_{m-1} \cup \{(x,y)\}}[L(x, y, f_{Z_{m-1}}(x))]. \tag{13}$$

The leave-one-out error, on the other hand, can be restated as

$$\mathbf{E}_{Z_m}[R_{LOO}(Z_m)] = \frac{1}{m}\sum_{i=1}^{m}\mathbf{E}_{Z_m}[L(x_i,y_i,f_{Z_m^i}(x_i))] \tag{14}$$

$$= \mathbf{E}_{Z_{m-1}\cup\{(x_m,y_m)\}}[L(x_m,y_m,f_{Z_{m-1}}(x_m))]. \tag{15}$$

Here we use the fact that expectation and summation can be interchanged. In addition, a permutation argument shows that all terms under the sum have to be equal, hence we can replace the average by one of the terms. Finally, if we rename (x_m,y_m) by (x,y), then (17) becomes identical to the rhs of (16) which proves the theorem.

■

This demonstrates that the leave-one-out error is a sensible quantity to use. We are short, however, of another key ingredient required in the use of this method when bounding the error of an estimator; we need a bound on the variance of $R_{LOO}(Z)$. While general results exist, which show that the leave-one-out estimator is not a worse estimate than the estimate based on the empirical error (see Kearns for example, who shows that at least the rate is not worse), we would expect that, on the contrary, the leave-one-out error is much more reliable than the empirical risk.

Below we state a result which is a slight improvement on Theorem 2 and which uses the same concentration of measure techniques used above.

Theorem 5 (Tail Bound for Leave-One-Out Estimators). *Denote by A a β-stable algorithm (for training set of size $m-1$) with the additional requirement that $0 \leq A(Z) \leq M$ for all $z \in X \times Y$ and for all training samples $Z \subset X \times Y$. Then we have:*

$$P\{|R_{LOO}(Z) - \boldsymbol{E}_Z[R_{LOO}(Z)]| > \varepsilon\} \leq 2\exp\Big(-\frac{2m\varepsilon^2}{(m\beta+M)^2}\Big). \tag{16}$$

Proof. The proof is very similar to that of Theorem 2 and uses Theorem 1. All we must do is show that $R_{LOO}(Z)$ does not change by too much if we replace one of the patterns in Z by a different pattern. This means that, for $Z^i := Z\backslash z_i \cup \{z\}$ (where $z := (x,y)$), we have to determine a constant c_0 such that

$$|R_{LOO}(Z) - R_{LOO}(Z^i)| \leq c_0 \text{ for all } i. \tag{17}$$

In the following we denote by f_Z^j (and $f_{Z_i}^j$ respectively) the estimate obtained when leaving out the j^{th} pattern. By direct computation

$$\begin{aligned}|R_{LOO}(Z) - R_{LOO}(Z^i)| \leq &\left[\frac{1}{m}\sum_{j\neq i}|L(x_j,y_j,f_Z^j(x_j)) - L(x_j,y_j,f_{Z^i}^j(x_j))|\right] + \\ &\frac{1}{m}|L(x_i,y_i,f_Z^i(x_i)) - L(x_i,y_i,f_{Z^i}^i(x_i))| \\ \leq &\frac{1}{m}\sum_{j\neq i}\beta + \frac{M}{m} < \beta + \frac{M}{m}.\end{aligned}$$

In the last inequality we use the fact that we have a β-stable algorithm, hence the individual summands are bounded by β. In addition, the loss at arbitrary locations is bounded from above by M (and by 0 from below), hence two losses may not differ by more than M overall. This shows that $c_o \leq \beta + \frac{M}{m}$. Using Theorem proves the bound. □

We may use the values of β computed for minimizers of the regularized risk functional (Theorem3)) in order to obtain practical bounds. The current result is an improvement on the confidence bounds available for minimizers of the regularized risk functional (there is no dependency on β in the confidence bound and the constants in the exponential term are slightly better). One would, however, suspect that much better bounds should be possible.

In particular, rather than bounding each individual term in the proof of Theorem3 by $\frac{\beta}{m}$, it should be possible to take advantage of averaging effects and, thus, replace the overall bound β by $\frac{\beta}{\sqrt{m}}$ for example. It is an open question whether such a bound can be obtained.

Also note that Theorem 5 only applies to Lipschitz continuous, convex loss functions. This means that we cannot use the bound in the case of classification, since the loss function is discontinuous (we have 0-1 loss). Still, the leave-oneout error turns out to be currently the most reliable estimator of the expected error available. Hence, despite some lack of theoretical justification, one should consider it a method of choice when performing model selection for kernel methods.

This brings us to another problem; how should we compute the leave-one-out error efficiently, without running a training algorithm m times? We must find approximations or good upper bounds for $R_{LOO}(Z)$ which are cheap to compute.

3 Entrophy and Covering Numbers

3.1 Definitions of Entrophy and Covering Numbers

Despite its improvement over the original definition, the fat shattering dimension is still a fairly crude summary of the capacity of the class of functions under consideration. Covering and entropy numbers can be used to derived more finely grained capacity measures. We begin with some definitions.

Recall that an ε-cover of a set M in E is a set of points in E such that the union of all ε-balls around these points contains M.

The *ε-covering number of $\mathscr{F}$ with respect to the metric d*, denoted $\mathscr{N}(\varepsilon, \mathscr{F}, d)$, is the smallest number of elements of an ε-cover for $\mathscr{F}$ using the metric d. Typically, $\mathscr{F}$ will be the class of functions under consideration. Moreover, d will be the metric induced by the values of $f \in \mathscr{F}$ on some data $X = \{x_1, \cdots, x_m\}$, such as the l_m^∞ metric. We denote this quantity by $l_m^\infty(X)$. For $\varepsilon = 1$ we recover the (scale less) definition of the covering number.

To avoid some of the technical difficulties, that come with this dependency on X, one usually takes the supremum of $\mathscr{N}(\varepsilon,\mathscr{F},l_m^\infty(X))$ with respect to X. This quantity will be called the ε-*growth function* of the function class $\mathscr{F}$. Formally we have

$$\mathscr{N}^m(\varepsilon,\mathscr{F}) := \sup_{x_1,\ldots,x_m\in X} \mathscr{N}(\varepsilon,\mathscr{F},l_\infty^X), \tag{18}$$

where $\mathscr{N}(\varepsilon,\mathscr{F},l_\infty^X)$ is the ε-covering number of $\mathscr{F}$ with respect to l_∞^X. Most generalization error bounds can be expressed in terms of $\mathscr{N}^m(\varepsilon,\mathscr{F})$. An example (Theorem 6) is given in the following section.

Covering numbers and the growth function are inherently discrete quantities. The functional inverse of $\mathscr{N}^m(\varepsilon,\mathscr{F})$, referred to as the entropy number, however, is more amenable to our analysis.The n^{th} *entropy number of a set* $M \subset E$, for $n \in N$, is given by

$$\varepsilon_n(M) := \inf\left\{\varepsilon > 0 \middle| \text{there exists an } \varepsilon\text{-cover for } M \text{ in } E \text{ containing } n \text{ or fewer points}\right\} \tag{19}$$

Since we are dealing with linear function classes, we will introduce the notion of entropy numbers of operators and represent the possible function values that these linear function classes can assume on the data as images of linear operators.

For this purpose we need to introduce the notion of entropy numbers of operators. Denote by E,G Banach spaces and by $\mathscr{L}(E,G)$ the space of linear operators from E into G. The *entropy numbers of an operator* $T \in \mathscr{L}(E,G)$ are defined as

$$\varepsilon_n(T) := \varepsilon_n(T(U_E)). \tag{20}$$

Note that $\varepsilon_1(T) - ||T||$, and that $\varepsilon_n(T)$ is well-defined for all $n \in N$ precisely if T is bounded. Moreover, $lim_{n\to\infty}\varepsilon_n(T) = 0$ if and only if T is compact; that is, if $T(U_E)$ is precompact.

A set is called *precompact* if its closure is compact. A set is called *compact* if every sequence in S has a subsequence that converges to an element also contained in S.

The *dyadic entropy numbers of an operator* are defined as

$$e_n(T) := \varepsilon_{2^{n-1}}(T), n \in N; \tag{21}$$

similarly, the dyadic entropy numbers of a set are defined from its entropy numbers. A beautiful introduction to entropy numbers of operators is given in a book by Carl and Stephani.

3.2 Generalization Bounds via Uniform Convergence

Let $E_m[f] := \frac{1}{m}\sum_{i=1}^m f(x_i)$ denote the *empirical mean* of f on the sample $x_1,\ldots,x_m$.

Theorem 6 (Alon, Ben-David, Cesa-Bianchi, and Haussler, 1997). *Let $\mathscr{F}$ be a class of functions from X into $[0,1]$. For all $\varepsilon > 0$, and all $m \geq \frac{2}{\varepsilon^2}$,*

$$P\left\{\sup_{f\in\mathscr{F}}|\boldsymbol{E}_m[f]-\boldsymbol{E}[f]|>\varepsilon\right\} \leq 12m\cdot\boldsymbol{E}\Big[\mathscr{N}\Big(\frac{\varepsilon}{6},\mathscr{F},l_\infty^{\bar{X}}\Big)\Big]\exp\Big(-\frac{\varepsilon^2 m}{36}\Big), \tag{22}$$

where the P on the left hand side denotes the probability w.r.t. the sample $x_1,\ldots,x_m$ drawn i.i.d. from the underlying distribution, and $\boldsymbol{E}$ the expectation w.r.t. a second sample $\bar{X} = (\bar{x}_1^\top,\ldots,\bar{x}_{2m}^\top)$, also drawn iid from the underlying distribution.

In order to use this result one usually makes use of the fact that, for any P,

$$\mathbf{E}_{\bar{X}}\big[\mathscr{N}(\varepsilon,\mathscr{F},l_\infty^m(\bar{X}))\big] \leq \mathscr{N}^m(\varepsilon,\mathscr{F}). \tag{23}$$

An alternative is to exploit the fact that $\mathscr{N}(\varepsilon,\mathscr{F},l_\infty^m(\bar{X}))$ is a concentrated random variable and measure $\mathscr{N}$ on the actual training set. Theorem 6 in conjunction with (23) can be used to give a generalization error result by applying it to the loss-function induced class.

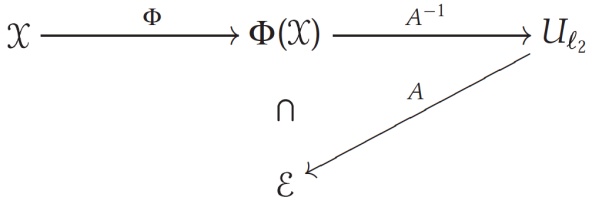
3.3 Entropy Numbers for Kernel Machines

Proposition 1 (Mapping Φ into l_2). *Let S be diagonal map*

$$\begin{aligned} S &: R^N \to R^N \\ &(x_j)_j \mapsto S(x_j)_j = (s_j x_j)_j, \end{aligned} \tag{24}$$

where $(s_j)_j \in R^N$. If $(s_j\sqrt{l_j})_j \in l_2$, then S maps $\Phi(X)$ into a ball centered at the origin whose radius is $R = ||(s_j\sqrt{l_j})_j||$.

As a consequence of Proposition 1, we can construct a mapping A from the unit ball in l_2 to an ellipsoid $\mathscr{E}$ such that $\Phi(X) \subset \mathscr{E}$, as in the following diagram:



The operator A will be useful for computing the entropy numbers of concatenations of operators. Knowing the inverse will allow us to compute the forward operator, and that can be used to bound the covering numbers of the class of functions, as shown in the next subsection.

Define

$$R := \left|\left|(s_j\sqrt{l_j})_j\right|\right|_{l_2} \tag{25}$$

From Proposition 1 it is clear that we may use

$$A = RS^{-1} = \left|\left|(s_j\sqrt{l_j})_j\right|\right|_{l_2} S^{-1} \tag{26}$$

We call such scaling (inverse) operators *admissible*. The next step is to compute the entropy numbers of the operator A and use this to obtain bounds on the entropy numbers for kernel machines such as SVMs.

The functions that an SV machine generates can be expressed as $x \mapsto \langle w, \Phi(x)\rangle + b$, where $w, \Phi(x) \in \mathscr{H}$ and $b \in R$. For now we consider the simplified class

$$\mathscr{F}_A := \{x \mapsto \langle w, \Phi(x)\rangle | x \in X, ||w|| \leq \Lambda\} \subseteq R^{\mathscr{H}}. \tag{27}$$

What we seek are the l_∞^m covering numbers for the class $\mathscr{F}_\Lambda$ induced by the kernel in terms of the parameter Λ. This is the inverse of the size of the margin in feature space, or, equivalently, the size of the weight SV Classes vector in feature space as defined by the dot product in $\mathscr{H}$. We call such hypothesis classes with a length constraint on the weight vectors in feature space *SV classes*. Let T be the operator $T = S_{\Phi(X)\Lambda}$ where $\Lambda \in R^+$, and define the operator $S_{\Phi(X)}$ by

$$\begin{aligned} &S_{\Phi(X)} : l_2 \to l_\infty^m \\ &S_{\Phi(X)} : w \mapsto (\langle \Phi(x_1), w\rangle, \ldots, \langle \Phi(x_m), w\rangle) \end{aligned} \tag{28}$$

The following theorem is useful in computing entropy numbers in terms of T and A.

For the next theorem:

$$\begin{aligned} &\Phi : X \to l_2^{N_{\mathscr{H}}} \\ &\quad x \mapsto (\sqrt{\lambda_j}\phi_j(x))_{j=1,\ldots,N_{\mathscr{H}}}, \end{aligned} \tag{29}$$

for almost all $x \in X$.

Theorem 7 (Bounds for SV classes). *Let k be a Mercer kernel and let $T := S_{\Phi(X)}\Lambda$ where $\Lambda \in \mathbb{R}^+$. Let A be defined by (30). Then the entropy numbers of T satisfy the following inequalities, for $n > 1$;*

$$\varepsilon_n(T) \leq c||A||\Lambda \log_2^{-1/2} n \log_2^{1/2}(1 + \frac{m}{\log_2 n}) \tag{30}$$

$$\varepsilon_n(T) \leq 6\Lambda\varepsilon_n(A) \tag{31}$$

$$\varepsilon_n(T) \leq 6c\Lambda \log_2^{-1/2} n \log_2^{1/2}(1 + \frac{m}{\log_2 n})\varepsilon_t(A). \tag{32}$$