E0 370 Statistical Learning Theory **Lecture 8 (Sep 8, 2011)**

Algorithmic Stability

Lecturer: Shivani Agarwal *Scribe: Harish Guruprasad*

1 Introduction

In the last few lectures we have seen a number of different generalization error bounds for learning algorithms, using notions such as the growth function and VC dimension; covering numbers, pseudo-dimension, and fat-shattering dimension; margins; and Rademacher averages. While these bounds are different in nature and apply in different contexts, a unifying factor that they all share is that that they hold uniformly for *all* functions in some fixed function class, not just for the function selected by the learning algorithm. In other words, the bounds hold for any algorithm that picks a function from the given class, no matter how the algorithm picks this function.

In this lecture we will see an alternative approach for obtaining generalization error bounds that takes into account the process by which a learning algorithm selects a function – in particular, we will see how to obtain bounds that apply to algorithms with good *stability* properties.

2 Stability

In general, an algorithm is *stable* if a small change in its input does not produce a drastic change in its output. A learning algorithm can be viewed as taking as input a training sample $S \in \bigcup_{m=1}^{\infty}(\mathcal{X} \times \mathcal{Y})^m$ and returning as output a function $f_S : \mathcal{X} \to \widehat{\mathcal{Y}}$. One can define various notions of stability for such an algorithm; we will consider two such notions below that are defined with respect to changes in the input consisting of replacing a single example in the training sample with a new example.

Notation. For a training sample $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$ and an example (x_i', y_i'), we will denote by $S^{i:(x_i', y_i')} = ((x_1, y_1), \ldots, (x_i', y_i'), \ldots, (x_m, y_m))$ the training sample obtained by replacing the ith example in S with (x_i', y_i'); we will sometimes abbreviate this as S^i when the replacement example is clear from context.

Definition. Let $\mathcal{Y}, \widehat{\mathcal{Y}} \subseteq \mathbb{R}$, and let $\mathcal{A}$ be a symmetric[1] algorithm that given a training sample $S \in (\mathcal{X} \times \mathcal{Y})^m$ as input, returns as output a function $f_S : \mathcal{X} \to \widehat{\mathcal{Y}}$. We say $\mathcal{A}$ has *(uniform, replacement) score stability* $\nu : \mathbb{N} \to [0, \infty)$ if $\forall m \in \mathbb{N}, i \in [m], S \in (\mathcal{X} \times \mathcal{Y})^m, (x_i', y_i') \in (\mathcal{X} \times \mathcal{Y}), x \in \mathcal{X}$,[2]

$$|f_S(x) - f_{S^i}(x)| \;\leq\; \nu(m)\,. \tag{1}$$

Let $\ell : \mathcal{Y} \times \widehat{\mathcal{Y}} \to [0, \infty)$ be a loss function. We say $\mathcal{A}$ has *(uniform, replacement) loss stability* $\beta : \mathbb{N} \to [0, \infty)$ with respect to ℓ if $\forall m \in \mathbb{N}, i \in [m], S \in (\mathcal{X} \times \mathcal{Y})^m, (x_i', y_i') \in \mathcal{X} \times \mathcal{Y}, (x, y) \in \mathcal{X} \times \mathcal{Y}$,

$$\Big|\ell(y, f_S(x)) - \ell(y, f_{S^i}(x))\Big| \;\leq\; \beta(m)\,. \tag{2}$$

[1] A symmetric learning algorithm $\mathcal{A} : \bigcup_{m=1}^{\infty}(\mathcal{X} \times \mathcal{Y})^m \to \mathcal{F}$ is one that does not depend on the order of the training examples, i.e. for all $m \in \mathbb{N}$, $S \in (\mathcal{X} \times \mathcal{Y})^m$, and permutations $\sigma \in S_m$, satisfies $\mathcal{A}(S) = \mathcal{A}(S_\sigma)$, where S_σ is the sample obtained by permuting the examples in S according to σ.

[2] The term uniform here refers to the fact that the bound is required to hold for all training samples S and replacement examples (x_i', y_i'); the term replacement refers to the fact that the small changes considered to the input involve replacement of an example in the training sample with another. Other notions of stability that relax/extend these requirements are also possible; we will mention some of these briefly later.

3 Generalization Error Bounds in Terms of Stability

Theorem 3.1. Let $\mathcal{Y}, \widehat{\mathcal{Y}} \subseteq \mathbb{R}$, and $\mathcal{A}$ a symmetric learning algorithm that given a training sample $S \in (\mathcal{X} \times \mathcal{Y})^m$, outputs a function $f_S : \mathcal{X} \to \widehat{\mathcal{Y}}$. Let $\ell : \mathcal{Y} \times \widehat{\mathcal{Y}} \to [0, B]$, and let $\mathcal{A}$ have loss stability β w.r.t. ℓ. Let D be any distribution on $\mathcal{X} \times \mathcal{Y}$ and let $0 < \delta < 1$. Then with probability at least $1 - \delta$ (over $S \sim D^m$),

$$\mathrm{er}_D^\ell[f_S] \;\leq\; \mathrm{er}_S^\ell[f_S] + \beta(m) + \Big(2m\beta(m) + B\Big)\sqrt{\frac{\ln(\frac{1}{\delta})}{2m}}\,.$$

Proof. Define $\phi : (\mathcal{X} \times \mathcal{Y})^m \to \mathbb{R}$ as

$$\phi(S) = \mathrm{er}_D^\ell[f_S] - \mathrm{er}_S^\ell[f_S]\,.$$

Then $\forall S, k, (x'_k, y'_k)$:

$$\begin{aligned}
\Big|\phi(S) - \phi(S^k)\Big| &\leq \Big|\mathrm{er}_D^\ell[f_S] - \mathrm{er}_D^\ell[f_{S^k}]\Big| + \Big|\mathrm{er}_S^\ell[f_S] - \mathrm{er}_{S^k}^\ell[f_{S^k}]\Big| && (3)\\
&= \Big|\mathbf{E}_{(x,y)\sim D}\left[\ell(y, f_S(x)) - \ell(y, f_{S^k}(x))\right]\Big| + \\
&\quad \left|\frac{1}{m}\sum_{i\neq k}\Big(\ell(y_i, f_S(x_i)) - \ell(y_i, f_{S^k}(x_i))\Big) + \frac{1}{m}\Big(\ell(y_k, f_S(x_k)) - \ell(y'_k, f_{S^k}(x'_k))\Big)\right| && (4)\\
&\leq \beta(m) + \frac{m-1}{m}\beta(m) + \frac{B}{m} && (5)\\
&\leq 2\beta(m) + \frac{B}{m}\,. && (6)
\end{aligned}$$

Therefore by McDiarmid's inequality (see Lecture 7 notes, Theorem 2.1), we have

$$\mathbf{P}_{S\sim D^m}\Big(\phi(S) - \mathbf{E}_{S\sim D^m}[\phi(S)] \geq \epsilon\Big) \;\leq\; e^{-2\epsilon^2/\left(m\left(2\beta(m) + \frac{B}{m}\right)^2\right)}\,. \qquad (7)$$

Rewriting the above, we have with probability at least $1 - \delta$ (over $S \sim D^m$),

$$\mathrm{er}_D^\ell[f_S] - \mathrm{er}_S^\ell[f_S] \;\leq\; \mathbf{E}_{S\sim D^m}\Big[\mathrm{er}_D^\ell[f_S] - \mathrm{er}_S^\ell[f_S]\Big] + (2m\beta(m) + B)\sqrt{\frac{ln(\frac{1}{\delta})}{2m}}\,. \qquad (8)$$

All that's left is to bound the expectation above:

$$\begin{aligned}
\mathbf{E}_{S\sim D^m}\left[\mathrm{er}_D^\ell[f_S] - \mathrm{er}_S^\ell[f_S]\right] &= \mathbf{E}_{S\sim D^m}\Big[\mathbf{E}_{(x,y)\sim D}\left[\ell(y, f_S(x))\right] - \frac{1}{m}\sum_{i=1}^m \ell(y_i, f_S(x_i))\Big] && (9)\\
&= \mathbf{E}_{(S,(x,y))\sim D^m\times D}\Big[\ell(y, f_S(x)) - \frac{1}{m}\sum_{i=1}^m \ell(y, f_{S^{i:(x,y)}}(x))\Big] && (10)\\
&= \mathbf{E}_{(S,(x,y))\sim D^m\times D}\Big[\ell(y, f_S(x)) - \ell(y, f_{S^{1:(x,y)}}(x))\Big] \quad \text{(by symmetry)} && (11)\\
&\leq \beta(m)\,. && (12)
\end{aligned}$$

Combining with the above gives the desired result. □

A few observations:

1. Unlike the uniform bounds we have seen previously, the above bound holds specifically for the function f_S learned by the algorithm.
2. Need $\beta(m) = o\left(\frac{1}{\sqrt{m}}\right)$ for the above bound to be useful.
3. For binary classification problems, cannot have non-trivial stability w.r.t. $\ell_{0\text{-}1}$ directly (why?), but if an algorithm has good stability w.r.t. some loss ℓ with $\ell \geq \ell_{0\text{-}1}$, then one can use the above result to obtain a high confidence bound on $\mathrm{er}_D^\ell[f_S]$, which in turn yields a bound on $\mathrm{er}_D^{0\text{-}1}[f_S]$.
4. If an algorithm $\mathcal{A}$ has score stability ν, then for any loss ℓ that is L-Lipschitz in its second argument, $\mathcal{A}$ has loss stability $\beta = L\nu$ w.r.t. ℓ.

4 Regularization Algorithms in an RKHS

We first briefly review some background material on reproducing kernel Hilbert spaces (RKHSs), and then discuss stability properties of kernel-based algorithms such as SVMs that learn a function using regularization in an RKHS.

4.1 Reproducing Kernel Hilbert Spaces

Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a symmetric, positive definite kernel function.[3,4] For each $x \in \mathcal{X}$, define $K_x : \mathcal{X} \to \mathbb{R}$ as $K_x(y) = K(x, y)$. Let

$$\mathcal{F}_0 = \left\{ f : \mathcal{X} \to \mathbb{R} \;\middle|\; f(x) = \sum_{i=1}^{m} \alpha_i K_{x_i}(x) \text{ for some } m \in \mathbb{N}, x_i \in \mathcal{X}, \alpha_i \in \mathbb{R} \right\}.$$

Define an inner product on $\mathcal{F}_0$ as

$$\left\langle \sum_{i=1}^{m} \alpha_i K_{x_i}, \sum_{j=1}^{n} \beta_j K_{y_j} \right\rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_i \beta_j K(x_i, y_j).$$

Let $\mathcal{F}_K$ be the completion of $\mathcal{F}_0$ w.r.t. the metric defined by the norm induced by the above inner product.[5] Then the function class $\mathcal{F}_K$ is called the *reproducing kernel Hilbert space* (RKHS) associated with the kernel function K. As the name suggests, any RKHS $\mathcal{F}_K$ forms a Hilbert space[6]; in addition, we have that for any $f \in \mathcal{F}_K$ and $x \in \mathcal{X}$,

$$\langle f, K_x \rangle = \left\langle \sum_i \alpha_i K_{x_i}, K_x \right\rangle = \sum_i \alpha_i K_{x_i}(x) = f(x).$$

This property is known as the *reproducing property* of $\mathcal{F}_K$.[7]

It is worth noting that the classifier learned by the SVM algorithm using a kernel function K has the form $h_S(x) = \text{sign}(f_S(x) + b)$ for some $f_S \in \mathcal{F}_K$ (or if trained without the bias parameter b, the classifier is simply $h_S(x) = \text{sign}(f_S(x))$ for some $f_S \in \mathcal{F}_K$); this follows directly from the fact that f_S is represented as $f_S(x) = \sum_{i=1}^{m} \alpha_i y_i K(x_i, x)$, where $S = ((x_1, y_1), \ldots, (x_m, y_m))$ is the training sample and α_i are the optimal values of the dual variables (see Lecture 2 notes).

We also mention the following well-known result:

Theorem 4.1 (Representer Theorem)**.** Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a symmetric, positive definite kernel function, and let $\mathcal{Y} \subseteq \mathbb{R}$. Let $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$. For any loss function $\ell : \mathcal{Y} \times \mathbb{R} \to [0, \infty)$, any solution f_S to the optimization problem

$$\min_{f \in \mathcal{F}_K} \frac{1}{m} \sum_{i=1}^{m} \ell(y_i, f(x_i)) + \frac{\lambda}{2} \|f\|_K^2$$

(where $\lambda > 0$) can be expressed as a kernel expansion on the points in S: $f_S(x) = \sum_{i=1}^{m} \alpha_i K(x_i, x)$ for some $\alpha_i \in \mathbb{R}$.

---

[3] Symmetry: $K(x, y) = K(y, x)\ \forall x, y \in \mathcal{X}$.

[4] Positive-definiteness: $\sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j K(x_i, x_j) > 0\ \forall m \in \mathbb{N}, x_1^m = (x_1, \ldots, x_m) \in \mathbb{R}^n, \boldsymbol{\alpha} \in \mathbb{R}^m, \boldsymbol{\alpha} \neq \mathbf{0}$.

[5] A metric space is said to be complete if every Cauchy sequence in the space converges to a limit in the space; any metric space can be completed by adding the limit points of all Cauchy sequences in the space to it. Here convergence of functions in $\mathcal{F}_K$ is with respect to the metric defined by $\|f - g\|_K = \sqrt{\langle f - g, f - g \rangle}$.

[6] A Hilbert space is simply an inner product space (vector space with an inner product) that is complete with respect to the metric defined by the associated inner product (see previous footnote).

[7] In general, a class of real-valued functions $\mathcal{F} \in \mathbb{R}^{\mathcal{X}}$ forms an RKHS if $\mathcal{F}$ is a Hilbert space and if there exists a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that (1) K satisfies the reproducing property in $\mathcal{F}$: $\langle f(\cdot), K(\cdot, x) \rangle = f(x)\ \forall f \in \mathcal{F}, x \in \mathcal{X}$, and (2) $\mathcal{F}$ is the completion of the span of $\{K(\cdot, x) \mid x \in \mathcal{X}\}$.

4.2 Stability of RKHS Regularization Algorithms

Theorem 4.2. Let $\mathcal{F}_K$ be an RKHS with kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $K(x,x) \leq \kappa^2 < \infty \; \forall x \in \mathcal{X}$. Let $\mathcal{Y} \subseteq \mathbb{R}$ and $\ell : \mathcal{Y} \times \mathbb{R} \to [0,\infty)$ be convex and L'-Lipschitz in its second argument. Let $\lambda > 0$, and let $\mathcal{A}$ be a symmetric algorithm that given a training sample S returns a function $f_S \in \mathcal{F}_K$ such that

$$f_S \;=\; \arg\min_{f \in \mathcal{F}_K} \mathrm{er}_S^\ell[f] + \frac{\lambda}{2}\|f\|_K^2 \,.$$

Then $\mathcal{A}$ has score stability

$$\nu(m) \;=\; \frac{4L'\kappa^2}{\lambda m} \,.$$

Proof. Let $S = ((x_1,y_1),\ldots,(x_m,y_m)) \in (\mathcal{X}\times\mathcal{Y})^m$, $i \in [m]$, $(x_i',y_i') \in \mathcal{X}\times\mathcal{Y}$. For brevity, let $f \equiv f_S$, $f^i \equiv f_{S^i}$; and let $\Delta f = f^i - f$. Our goal is to show $|\Delta f(x)| \leq \frac{4L'\kappa^2}{\lambda m} \; \forall x \in \mathcal{X}$.

Recall that any convex function $\phi : U \to \mathbb{R}$ satisfies the following for all $u, v \in U$ and $t \in [0,1]$:

$$\phi\big(u + t(v-u)\big) - \phi(u) \;\leq\; t\Big(\phi(v) - \phi(u)\Big) \,.$$

Since ℓ is convex in its second argument, we have that $\mathrm{er}_S^\ell[f]$ is convex in f. Therefore we have $\forall t \in [0,1]$:

$$\begin{aligned}
\mathrm{er}_S^\ell[f + t(f^i - f)] - \mathrm{er}_S^\ell[f] &\leq t\left(\mathrm{er}_S^\ell[f] - \mathrm{er}_S^\ell[f^i]\right) && (13)\\
\mathrm{er}_S^\ell[f^i + t(f - f^i)] - \mathrm{er}_S^\ell[f^i] &\leq t\left(\mathrm{er}_S^\ell[f^i] - \mathrm{er}_S^\ell[f]\right) \,. && (14)
\end{aligned}$$

Adding the above gives

$$\mathrm{er}_S^\ell[f + t\Delta f] + \mathrm{er}_S^\ell[f^i - t\Delta f] \;\leq\; \mathrm{er}_S^\ell[f] + \mathrm{er}_S^\ell[f^i] \,. \qquad (15)$$

Now since $\mathcal{F}_K$ is a vector space, we have $f + t\Delta f \in \mathcal{F}_K$, $f^i - t\Delta f \in \mathcal{F}_K$. Therefore, since f and f^i minimize over all functions in $\mathcal{F}_K$ the regularized empirical ℓ-error w.r.t. S and S^i, respectively, we have

$$\begin{aligned}
\mathrm{er}_S^\ell[f] + \frac{\lambda}{2}\|f\|_K^2 &\leq \mathrm{er}_S^\ell[f + t\Delta f] + \frac{\lambda}{2}\|f + t\Delta f\|_K^2 && (16)\\
\mathrm{er}_{S^i}^\ell[f^i] + \frac{\lambda}{2}\|f^i\|_K^2 &\leq \mathrm{er}_{S^i}^\ell[f^i - t\Delta f] + \frac{\lambda}{2}\|f^i - t\Delta f\|_K^2 \,. && (17)
\end{aligned}$$

Adding Eqs. (15-17) then yields

$$\begin{aligned}
&\frac{\lambda}{2}\left(\|f\|_K^2 + \|f^i\|_K^2 - \|f + t\Delta f\|_K^2 - \|f^i - t\Delta f\|_K^2\right) && (18)\\
&\leq \left(\mathrm{er}_{S^i}^\ell[f^i - t\Delta f] - \mathrm{er}_S^\ell[f^i - t\Delta f]\right) + \left(\mathrm{er}_S^\ell[f^i] - \mathrm{er}_{S^i}^\ell[f^i]\right) && (19)\\
&= \frac{1}{m}\left(\ell(y_i', (f^i - t\Delta f)(x_i')) - \ell(y_i, (f^i - t\Delta f)(x_i))\right) + \frac{1}{m}\left(\ell(y_i, f^i(x_i)) - \ell(y_i', f^i(x_i'))\right) && (20)\\
&= \frac{1}{m}\left(\ell(y_i', (f^i - t\Delta f)(x_i')) - \ell(y_i', f^i(x_i'))\right) + \frac{1}{m}\left(\ell(y_i, f^i(x_i)) - \ell(y_i, (f^i - t\Delta f)(x_i))\right) && (21)\\
&\leq \frac{L'}{m}\left(|t\Delta f(x_i')| + |t\Delta f(x_i)|\right) \quad \text{(by Lipschitz property)} && (22)\\
&= \frac{tL'}{m}\left(|\Delta f(x_i')| + |\Delta f(x_i)|\right) && (23)\\
&\leq \frac{2tL'\kappa}{m}\|\Delta f\|_K \,, && (24)
\end{aligned}$$

where the last line follows by the reproducing property of $\mathcal{F}_K$, Cauchy-Schwartz inequality, and the fact that $K(x,x) \leq \kappa^2 \; \forall x \in \mathcal{X}$. Taking $t = 1/2$ then gives

$$\frac{\lambda}{4}\|\Delta f\|_K^2 \;\leq\; \frac{L'\kappa}{m}\|\Delta f\|_K \qquad (25)$$

which simplifies to

$$\|\Delta f\|_K \;\leq\; \frac{4L'\kappa}{\lambda m}\,. \tag{26}$$

Finally, using the reproducing property and Cauchy-Schwartz again, we get

$$|\Delta f(x)| \;=\; |\langle \Delta f, K_x\rangle| \;\leq\; \|\Delta f\|_K\sqrt{\langle K_x, K_x\rangle} \;\leq\; \frac{4L'\kappa^2}{\lambda m}\,. \tag{27}$$

The result follows. □

Stability of SVMs. To see how the above result can be applied to obtain a generalization error bound for the SVM algorithm, note that SVM satisfies the conditions of the theorem with $\ell = \ell_{\text{hinge}}$, with $L' = 1$. This gives that the SVM algorithm using a kernel function K with $K(x,x) \leq \kappa^2 < \infty\ \forall x$ has score stability

$$\nu(m) = \frac{4\kappa^2}{\lambda m}\,.$$

By observation 4 in Section 3, it follows that the SVM algorithm with kernel K as above then has loss stability $\frac{4\kappa^2}{\lambda m}$ w.r.t. ℓ_{hinge} (taking now $L = 1$); however since the hinge loss is not bounded, we cannot use this to obtain a generalization error bound from Theorem 3.1. Instead, consider the ramp loss $\ell_{\text{ramp}} = \ell_{\text{ramp}(1)}$ (see Lecture 6 notes), which is also 1-Lipschitz, so that the SVM algorithm as above has loss stability $\frac{4\kappa^2}{\lambda m}$ w.r.t. ℓ_{ramp} as well; since ℓ_{ramp} is bounded in $[0,1]$ and also forms an upper bound on $\ell_{0\text{-}1}$, we can then apply Theorem 3.1 to obtain that with probability at least $1-\delta$ over $S \sim D^m$, the function f_S learned by the SVM algorithm with kernel K as above satisfies

$$\text{er}_D^{0\text{-}1}[f_S] \;\leq\; \text{er}_S^{\text{ramp}}[f_S] + \frac{4\kappa^2}{\lambda m} + \Big(\frac{8\kappa^2}{\lambda} + B\Big)\sqrt{\frac{ln(\frac{1}{\delta})}{2m}}\,.$$

Note that this bound can be applied to the function learned by the SVM algorithm using *any* kernel function K for which $K(x,x)$ is bounded, including for example the Gaussian kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ given by $K(\mathbf{x},\mathbf{x}') = \exp(\frac{-\|\mathbf{x}-\mathbf{x}'\|_2^2}{2\sigma^2})$, for which $K(\mathbf{x},\mathbf{x}) \leq 1\ \forall \mathbf{x}$; the Gaussian kernel is known to induce an RKHS $\mathcal{F}_K$ for which the associated binary class $\text{sign}(\mathcal{F}_K)$ has infinite VC-dimension, and therefore VC-dimension bounds cannot be applied to the SVM with this kernel.

The same technique as above can be used to show stability (and generalization error bounds) for support vector regression (SVR), as well as other regularization-based algorithms; see [1] for more details.

5 Deletion Stability, Leave-one-out Error, and Other Extensions

The notions of stability defined above are in terms of changes to the training sample that consist of replacing one example; one can also define similar notions of stability in terms of changes consisting of *deleting* one example from the sample. Deletion stability clearly implies replacement stability, and is therefore a slightly stronger condition.

So far, we have used various forms of empirical error (average loss on the training sample, using different loss functions) to estimate or obtain bounds on the generalization error of a learned function. However the empirical error is not the only quantity that can be used to obtain such bounds; other quantities can also be appropriate. One such quantity is the *leave-one-out* error, which is obtained by training an algorithm $\mathcal{A}$ on m different subsamples $S^{\setminus i}, i = 1,\ldots,m$ of the training sample $S = ((x_1,y_1),\ldots,(x_m,y_m))$, where $S^{\setminus i}$ denotes the sample obtained by removing the i-th example from S, then testing each learned function $f_{S^{\setminus i}}$ on the left-out-example (x_i,y_i) and averaging the result:

$$\text{er}_{\text{loo}}[\mathcal{A};S] \;=\; \frac{1}{m}\sum_{i=1}^{m}\ell\big(y_i, f_{S^{\setminus i}}(x_i)\big)\,, \tag{28}$$

where $f_{S^{\setminus i}} = \mathcal{A}(S^{\setminus i})$. Deletion stability of $\mathcal{A}$ can also be used to obtain bounds on the generalization error $\mathrm{er}_D[f_S]$ of $f_S = \mathcal{A}(S)$ similar to those above but in terms of the leave-one-out error $\mathrm{er}_{\mathrm{loo}}[\mathcal{A}; S]$ rather than the empirical error $\mathrm{er}_S[f_S]$; see [1] for more details.

In certain cases, it is also helpful to consider weaker notions of stability, such as requiring the bounded change in the learned function to hold not necessarily uniformly over all training samples and changes (replacements/deletions), but only with high probability or in expectation over the samples/changes; this leads to distribution-dependent forms of stability [1, 2].

6 Next Lecture

In the next lecture we will consider bounding the generalization error of functions learned from a hierarchy of models (function classes) using model selection techniques.

References

[1] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[2] Samuel Kutin and Partha Niyogi. Almost-everywhere algorithmic stability and generalization error. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, 2002.