ized situation, we have m independent datasets of size n, composed of independent and identically distributed (i.i.d.) observations from the same distribution $p(x, y)$ on $\mathcal{X} \times \mathcal{Y}$. We obtain for each of them an estimator $\hat{f}_\lambda^{(j)}$, where $j \in \{1, \dots, m\}$ and λ is an associated hyperparameter specific to the learning procedure. Since we consider least-squares regression, the new predictor, $\hat{f}_\lambda^{\text{bag}}$, is simply the average of all $\hat{f}_\lambda^{(j)}$, $j = 1, \dots, m$.

If we denote $\text{bias}^{(j)}(x) = \mathbb{E}[\hat{f}_\lambda^{(j)}(x)] - f_*(x)$, and $\text{var}^{(j)}(x) = \text{var}\big[\hat{f}_\lambda^{(j)}(x)\big]$ (assuming that x is fixed and only taking expectations with respect to the data), then they are the same for all $j \in \{1, \dots, m\}$ and the bias of $\hat{f}_\lambda^{\text{bag}}$ is the same as the base bias for a single dataset (and thus so is the squared bias). At the same time, the variance is divided by m because the datasets are assumed to be independent.

Thus, in the bias/variance trade-off, the selected hyperparameter will typically select a higher variance (or equivalently lower bias) estimator than for $m = 1$. In the context of independent datasets, it is relevant to concatenate all m datasets into one large dataset with $N = nm$ observations and learn a single predictor with these: the generalization performance of the average of m predictors will often be the same as the one of the single predictor on the large dataset, but with potential computational benefits. We now give a few examples for regression (we consider binary classification in exercises 10.1 and 10.3).

The k-nearest neighbor regression. We consider the analysis from section 6.3.2 on prediction problems over $\mathcal{X} \subset \mathbb{R}^d$, where we showed in proposition 6.2 that the (squared) bias was upper-bounded by $8B^2\text{diam}(\mathcal{X})^2\big(\frac{2k}{n}\big)^{2/d}$ (for $d \geqslant 2$). At the same time, the variance was bounded by $\frac{\sigma^2}{k}$, where σ^2 is a bound on the noise variance on top of the target function f_*, while B is the Lipschitz constant of the target function. Thus, with m replications, we get an excess risk upper-bounded by

$$\frac{\sigma^2}{km} + 8B^2\text{diam}(\mathcal{X})^2\Big(\frac{2k}{n}\Big)^{2/d}.$$

When optimizing this bound with respect to k, we get that $k^{1+2/d} \propto \frac{n^{2/d}}{m}$, leading to $k \propto \frac{1}{m^{d/(2+d)}} n^{2/(2+d)}$. Compared to section 6.3.2, we obtain a smaller number of neighbors (which is consistent with favoring higher variance estimators). The overall excess risk ends up being proportional to $1/(mn)^{2/(d+2)}$, which is exactly the rate for a dataset of $N = mn$ observations.

Thus, dividing a dataset of N observations in m chunks of $n = N/m$ observations, estimating independently, and combining linearly does not lead to an overall improved statistical behavior compared to learning all at once. Still, it can have significant computational advantages when the m estimators can be computed in parallel (and totally independently). We thus obtain a distributed algorithm with the same worst-case predictive performance as for a single machine.

Note here that there is an upper bound on the number of replications (and thus the ability for parallelization) to get the same (optimal) rate, as we need k to be larger than 1, and thus, m cannot grow larger than $n^{2/d}$.

Exercise 10.1 (♦) *Consider k-nearest neighbor multicategory classification with a majority vote rule. Using the relationship between the quadratic loss and the 0–1 loss from section 4.1.4 to derive an upper bound on the expected risk, what is the corresponding optimal choice of m when using independent datasets?*

Ridge regression. Following the analysis from section 7.6.6, the variance of the ridge regression estimator was proportional to $\frac{\sigma^2}{n}\lambda^{-1/\alpha}$ and the bias proportional to $\lambda^{t/s}$ (see precise definitions in section 7.6.6). With m replications, we thus get an excess risk proportional to $\frac{\sigma^2}{nm}\lambda^{-1/\alpha} + \lambda^{t/s}$, and the averaged estimator behaves like having $N = nm$ observations (and the same regularization parameter). Again, with the proper choice of regularization parameter (lower λ than for the full dataset), there is no statistical advantage. Still, there may be a computational one, not only for parallel processing but also with a single machine (see exercise 10.2), since, as shown in section 7.4, the training time for ridge regression is superlinear in the number of observations with running-time complexities between $O(n^2)$ and $O(n^3)$ if no low-rank approximations are used.

Exercise 10.2 *Assuming that obtaining an estimator for ridge regression has running-time complexity $O(n^\beta)$ for $\beta \geqslant 1$ for n observations, what is the complexity of using a split of the data into m chunks? What is the optimal value of m?*

Exercise 10.3 (♦) *In the setup of this section with m independent datasets, consider an estimator $\hat{f}^{(j)} : \mathcal{X} \to \{-1, 1\}$ learned on the jth dataset for a binary classification problem, for $j \in \{1, \dots, m\}$, with $\hat{f}^{\text{bag}}(x) = \operatorname{sign}\big(\sum_{j=1}^m \hat{f}^{(j)}(x)\big)$ the majority vote classifier. Denoting $f_*(x) \in \{-1, 1\}$ the optimal prediction at $x \in \mathcal{X}$, as defined in section 2.2.3, and $\varepsilon(x) = \mathbb{E}[\hat{f}^{(j)}(x) f_*(x)]$ (which is the same for all j), show that we have $\mathbb{E}[1_{\hat{f}^{\text{bag}}(x) \neq f_*(x)}] \leqslant \exp\big(-\frac{m}{2}(\varepsilon(x))_+^2\big)$. If $\forall x \in \mathcal{X}$, $\varepsilon(x) \geqslant \eta > 0$, show that the expected excess risk is less than $\exp\big(-\frac{m}{2}\eta^2\big)$.*

Beyond independent datasets. Having independent datasets may not be possible, and one typically needs to artificially create such replicated datasets from a single one, which is precisely what bagging methods will do in section 10.1.2, with a reduced variance still, but this time with a potentially higher bias.

10.1.2 Bagging

We consider datasets $\mathcal{D}^{(b)}$, obtained with random weights $v_i^{(b)} \in \mathbb{R}_+$, $i = 1, \dots, n$. For the bootstrap,[1] we consider n samples from the original n data points with replacement, which correspond to integer weights $v_i^{(b)} \in \mathbb{N}$, $i = 1, \dots, n$, that sum to n. Such sets of weights are sampled independently m times. We study $m = \infty$ for simplicity; that is, infinitely many replications (in practice, the infinite m behavior can be achieved with moderate m's). Infinitely many bootstrap replications lead to a form of stabilization,

[1] See https://en.wikipedia.org/wiki/Bootstrapping_(statistics) and Efron and Tibshirani (1994) for an introduction to bootstrapping methods in statistics.

which is important for highly variable predictors (which usually imply a large estimation variance).

For linear estimators (in the definition of section 6.2.1; see also section 7.6.1) with the square loss, such as kernel ridge regression or local averaging, this leads to another linear estimator. Therefore, this provides alternative ways of regularizing, which typically may not provide a strong statistical gain over existing methods but provide a computational gain, in particular when each estimator is very efficient to compute (see related examples in section 10.2). Overall, as will be shown for 1-nearest-neighbor, bagging will reduce variance while increasing the bias, thus leading to trade-offs that are common in regularizing methods. See also the end of section 10.2 for a short description of "random forests," which is also partially based on bagging.

For simplicity, we will consider averaging estimators obtained by randomly selecting s observations from the n available ones, doing this many times (infinitely many for the analysis), and averaging the predictions.

Exercise 10.4 *Show that when sampling n elements with replacement from n items, the expected fraction of distinct items is $1-(1-1/n)^n$ and it tends to $1-1/e$ when n tends to infinity.*

One-nearest neighbor regression. We focus on the 1-nearest neighbor estimator where the strong effect of bagging is striking. The analysis in this subsection follows from Biau et al. (2010). The key observation is that if we denote as $(x_{(i)}(x), y_{(i)}(x))$ the pair of observations that is the ith-nearest neighbor of x from the dataset $x_1, \dots, x_n$ (ignoring ties), then we can write the bagged estimate as

$$\hat{f}(x) = \sum_{i=1}^{n} V_i y_{(i)}(x),$$

where the nonnegative weights V_i sum to 1 and *do not depend on* x. The weight V_i is the probability that the ith-nearest neighbor of x is the 1-nearest-neighbor of x in a uniform subsample of size s. We consider sampling without replacement and leave sampling with replacement as an exercise (see Biau et al., 2010, for more details). We assume that $s \geqslant 2$.

To select the ith-nearest neighbor as the 1-nearest-neighbor in a subsample, we need that the ith-nearest neighbor is selected but none of the closer neighbors, which leaves $s-1$ elements to choose among $n-i$ possibilities. This shows, that if $i > n-s+1$, then $V_i = 0$, while otherwise $V_i = \binom{n}{s}^{-1}\binom{n-i}{s-1}$, as the total number of subsets of size s is $\binom{n}{s}$, and there are $\binom{n-i}{s-1}$ relevant ones.

We can now use the reasoning from section 6.3.2. Since for any x, the weights given to each observation (once they are ordered in terms of distance to x) are $V_1, \dots, V_n$, the variance term is equal to $\sum_{i=1}^n V_i^2$. To obtain a bound, we note that for $i \leqslant n-s+1$,

$$V_i = \frac{s}{n-s+1}\frac{\prod_{j=0}^{s-2}(n-i-j)}{\prod_{j=0}^{s-2}(n-j)} = \frac{s}{n-s+1}\prod_{j=0}^{s-2}\Big(1-\frac{i}{n-j}\Big) \leqslant \frac{s}{n-s+1}\prod_{j=0}^{s-2}\Big(1-\frac{i}{n}\Big),$$

leading to, upper-bounding the sum by an integral,

$$\begin{aligned}\sum_{i=1}^n V_i^2 &\leqslant \frac{s^2}{(n-s+1)^2}\sum_{i=1}^n \big(1-\frac{i}{n}\big)^{2(s-1)} \leqslant \frac{ns^2}{(n-s+1)^2}\int_0^1 (1-t)^{2(s-1)}dt \\ &\leqslant \frac{ns^2}{(n-s+1)^2}\frac{1}{2s-1} \leqslant \frac{ns}{(n-s+1)^2} = \frac{s}{n}\frac{1}{(1+1/n-s/n)^2}.\end{aligned}$$

For the bias term, we need to bound $\sum_{i=1}^n V_i \cdot \mathbb{E}\big[\|x-x_{(i)}(x)\|^2\big]$, where the expectation is with respect to the data and the test point x. We note here that by definition of V_i, and conditioning on the data and x, this is B^2 multiplied by the expectation of the distance to the first nearest neighbor from a random sample of size s, and thus, for the ℓ_∞-norm on a subset $\mathcal{X}$ of $\mathbb{R}^d$, from lemma 6.1, less than $4B^2\mathrm{diam}(\mathcal{X})^2\frac{1}{s^{2/d}}$ if $d \geqslant 2$ (which we now assume).

Thus, the overall excess risk is less than

$$4B^2\mathrm{diam}(\mathcal{X})^2\frac{1}{s^{2/d}} + \frac{s}{n}\frac{1}{(1+1/n-s/n)^2},$$

which we can balance by choosing $s^{1+2/d} \propto n$, leading to the same performance as the k-nearest neighbor for a well-chosen k, but now with a bagged esimate.

In figure 10.1, simulations in one dimension are plotted, showing the regularizing effects of bagging; we see that when $s = n$ (no subsampling), we recover the 1-nearest neighbor estimate, and when s decreases, the variance indeed decreases while the bias increases.

10.2 Random Projections and Averaging

In section 10.1, we reweighted observations to be able to rerun the original algorithm. This can also be done through random projections of all observations. Such random projections can be performed in several ways: (1) for data in $\mathbb{R}^d$ by selecting s of the d variables; (2) still for data in $\mathbb{R}^d$, by projecting the data in a more general s-dimensional subspace; and (3) for kernel methods, using random features such as presented in section 7.4. Such random projections can also reduce the number of samples while keeping the dimension fixed (this will depend if the design matrix is left- or right-multiplied by a matrix of reduced size).

In this section, we consider random projections for ordinary least-squares (OLS), with the same notation as in chapter 3, with $y \in \mathbb{R}^n$ the response vector and $\Phi \in \mathbb{R}^{n\times d}$ the design matrix, in two settings:

- *Sketching*: Replacing $\min_{\theta\in\mathbb{R}^d} \|y-\Phi\theta\|_2^2$ by $\min_{\theta\in\mathbb{R}^d} \|Sy-S\Phi\theta\|_2^2$, where $S \in \mathbb{R}^{s\times n}$ is an i.i.d. Gaussian matrix (with independent zero mean and unit variance elements). This is an idealization of subsampling done in the previous section. Here, we typically have $n > s > d$ (more observations than the feature dimension), and

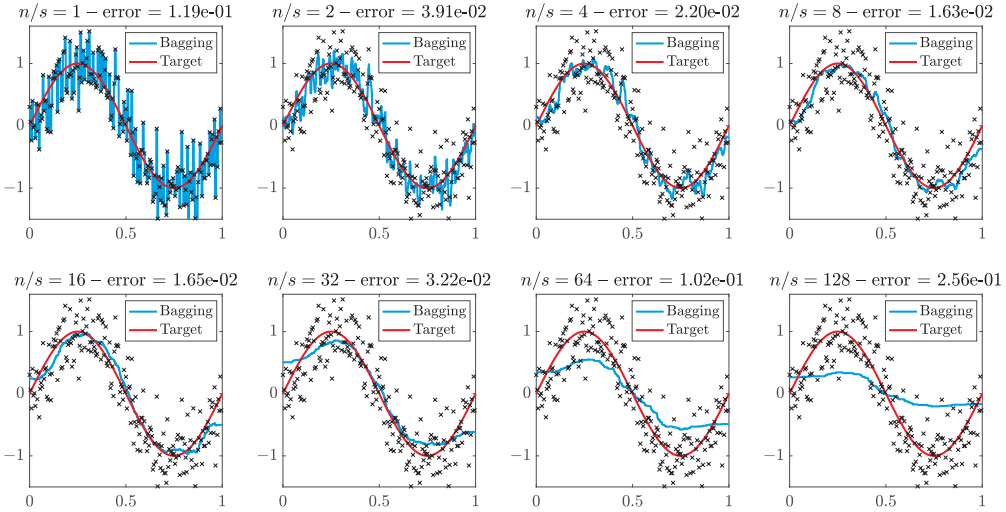

Figure 10.1. Subsampling estimates with $m = 20$ subsampled datasets, for varying subsampling ratios n/s, with an estimation of the testing error. When $n/s = 1$, we recover the 1-nearest neighbor classifier (which overfits), and when n/s grows, we get better fits until underfitting kicks in. Optimal testing error is obtained for $n/s = 8$.

one of the benefits of sketching is to be able to store a reduced representation of the data ($\mathbb{R}^{s\times d}$ instead of $\mathbb{R}^{n\times d}$).

- *Random projection*: Replacing $\min_{\theta\in\mathbb{R}^d} \|y - \Phi\theta\|_2^2$ by $\min_{\eta\in\mathbb{R}^s} \|y - \Phi S\eta\|_2^2$, where $S \in \mathbb{R}^{d\times s}$ is a more general sketching matrix. Here, we typically have $d > n > s$ (high-dimensional situation). The benefits of random projection are twofold: reduction in computation time and regularization. This corresponds to replacing the corresponding feature vectors $\varphi(x) \in \mathbb{R}^d$ by $S^\top\varphi(x) \in \mathbb{R}^s$. We will consider Gaussian matrices, but also subsampling matrices, and draw connections with kernel methods.

In the following sections, we study these precisely for the OLS framework (it could also be done for ridge regression). We first briefly mention a commonly used and related approach.

Random forests. A popular algorithm called "random forests" (Breiman, 2001) mixes both dimension reduction by projection and bagging: decision trees are learned on a bootstrapped sample of the data, while selecting a random subset of features at every splitting decision. This algorithm has nice properties (invariance to rescaling of the variables and robustness in high dimension due to the random feature selection) and can be extended in many ways. See Biau and Scornet (2016) for details.

10.2.1 Gaussian Sketching

Following section 3.3 on OLS, we consider a design matrix $\Phi \in \mathbb{R}^{n \times d}$ with rank d (i.e., $\Phi^\top \Phi \in \mathbb{R}^{d \times d}$ invertible), which implies $n \geqslant d$. We consider $s > d$ Gaussian random projections, with typically $s \leqslant n$, but this is not necessary in the analysis that follows.

The estimator $\hat{\theta}^{(j)}$ is obtained by using $S^{(j)} \in \mathbb{R}^{s \times n}$, with $j = 1, \dots, m$, where m denotes the number of replications. We then consider $\hat{\theta} = \frac{1}{m} \sum_{j=1}^{m} \hat{\theta}^{(j)}$. When $m = 1$, this is a single sketch.

We will consider the same fixed design assumptions as in section 3.5; that is, $y = \Phi\theta_* + \varepsilon$, where $\varepsilon \in \mathbb{R}^n$ has independent zero-mean components with variance σ^2, and $\theta_* \in \mathbb{R}^d$. Our goal is to compute the fixed design error $\frac{1}{n}\mathbb{E}_{\varepsilon,S}\|\Phi\hat{\theta} - \Phi\theta_*\|_2^2$, where we take expectations with respect to both the learning problem (in the fixed design setting, the noise vector ε) and the added randomization (the sketching matrices $S^{(j)}$, $j = 1, \dots, m$).

To compute this error, we first need to compute expectations and variances with respect to the random projections, assuming that ε is fixed.

We first introduce a representation tool that will allow simple expressions of all prediction vectors $S^{(j)}\Phi$. Since the Gaussian matrices $S^{(j)}$ are invariant by left and right multiplication by an orthogonal matrix, we can assume that the singular value decomposition (SVD) of $\Phi = UDV^\top$, where $V \in \mathbb{R}^{d \times d}$ is orthogonal (i.e., $V^\top V = VV^\top = I$), $D \in \mathbb{R}^{d \times d}$ is an invertible diagonal matrix, and $U \in \mathbb{R}^{n \times d}$ has orthonormal columns (i.e., $U^\top U = I$; remember than $n \geqslant d$), is such that $U = \binom{I}{0}$, and we can write $S^{(j)} = \begin{pmatrix} S_1^{(j)} & S_2^{(j)} \end{pmatrix}$, with $S_1^{(j)} \in \mathbb{R}^{s \times d}$ and $S_2^{(j)} \in \mathbb{R}^{s \times (n-d)}$. We can also split y as $y = \binom{y_1}{y_2}$ for $y_1 \in \mathbb{R}^d$ and $y_2 \in \mathbb{R}^{n-d}$.

We can write the normal equation that defines $\hat{\theta}^{(j)} \in \mathbb{R}^d$, for each $j \in \{1, \dots, m\}$ (i.e., $(\Phi^\top (S^{(j)})^\top S^{(j)} \Phi)\hat{\theta}^{(j)} = \Phi^\top (S^{(j)})^\top S^{(j)} y$), leading to the following closed-form estimators $\hat{\theta}^{(j)} = (\Phi^\top (S^{(j)})^\top S^{(j)} \Phi)^{-1} \Phi^\top (S^{(j)})^\top S^{(j)} y$.[2] Using the assumptions given previously regarding the SVD of Φ, we have $S^{(j)}\Phi = S_1^{(j)} D V^\top$. We can then expand the prediction vector in $\mathbb{R}^n$ as

$$
\begin{aligned}
\Phi\hat{\theta}^{(j)} &= \Phi(\Phi^\top (S^{(j)})^\top S^{(j)} \Phi)^{-1} \Phi^\top (S^{(j)})^\top S^{(j)} y \\
&= \binom{I}{0} D V^\top (V D (S_1^{(j)})^\top S_1^{(j)} D V^\top)^{-1} V D (S_1^{(j)})^\top S^{(j)} y \\
&= \binom{I}{0} ((S_1^{(j)})^\top S_1^{(j)})^{-1} (S_1^{(j)})^\top S^{(j)} y = \binom{I}{0} ((S_1^{(j)})^\top S_1^{(j)})^{-1} (S_1^{(j)})^\top (S_1^{(j)} y_1 + S_2^{(j)} y_2) \\
&= \begin{pmatrix} y_1 + ((S_1^{(j)})^\top S_1^{(j)})^{-1} (S_1^{(j)})^\top S_2^{(j)} y_2 \\ 0 \end{pmatrix}.
\end{aligned}
$$

Thus, since $\mathbb{E}[S_2^{(j)}] = 0$ and $S_2^{(j)}$ is independent of $S_1^{(j)}$, we get $\mathbb{E}_{S^{(j)}}\left[\Phi\hat{\theta}^{(j)}\right] = \binom{y_1}{0}$, which happens to be exactly the OLS estimator $\Phi\hat{\theta}_{\mathrm{OLS}} = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top y = \left(\begin{smallmatrix} I & 0 \\ 0 & 0 \end{smallmatrix}\right) y$. Moreover,

[2] If $s \geqslant d$, then $S^{(j)}\Phi$ has rank d almost surely, and thus $\hat{\theta}^{(j)}$ is uniquely defined.

we have the model $y = \Phi\theta_* + \varepsilon$ and, if we split ε as $\varepsilon = \binom{\varepsilon_1}{\varepsilon_2}$, we have $y = \binom{I}{0} DV^\top \theta_* + \varepsilon$, and thus $y_2 = \varepsilon_2$. We therefore get

$$\mathbb{E}_{S^{(j)}}\Big[\big\|\Phi\hat\theta^{(j)} - \mathbb{E}_{S^{(j)}}\Phi\hat\theta^{(j)}\big\|_2^2\Big] = \mathbb{E}_{S^{(j)}}\big[\|((S_1^{(j)})^\top S_1^{(j)})^{-1}(S_1^{(j)})^\top S_2^{(j)}\varepsilon_2\|_2^2\big].$$

Taking the expectation with respect to ε (through $\mathbb{E}[\varepsilon_2\varepsilon_2^\top] = \sigma^2 I$), using the trace trick, and using expectations for the Wishart and inverse Wishart distributions,[3] this leads to

$$\begin{aligned}
\mathbb{E}_{\varepsilon,S^{(j)}}\Big[\big\|\Phi\hat\theta^{(j)} - \mathbb{E}_{S^{(j)}}\Phi\hat\theta^{(j)}\big\|_2^2\Big] &= \sigma^2\mathbb{E}_{S^{(j)}}\big[\operatorname{tr}\big((S_2^{(j)})^\top S_1^{(j)}((S_1^{(j)})^\top S_1^{(j)})^{-2}(S_1^{(j)})^\top S_2^{(j)}\big)\big] \\
&= \sigma^2\mathbb{E}_{S^{(j)}}\big[\operatorname{tr}\big(S_2^{(j)}(S_2^{(j)})^\top S_1^{(j)}((S_1^{(j)})^\top S_1^{(j)})^{-2}(S_1^{(j)})^\top\big)\big] \\
&= (n-d)\sigma^2\mathbb{E}_{S_1^{(j)}}\big[\operatorname{tr}\big(((S_1^{(j)})^\top S_1^{(j)})^{-1}\big)\big] = \frac{d}{s-d-1}(n-d)\sigma^2.
\end{aligned}$$

We can now compute the overall expected generalization error:

$$\begin{aligned}
\frac{1}{n}\mathbb{E}_{\varepsilon,S^{(j)}}\Big[\Big\|\frac{1}{m}\sum_{j=1}^m \Phi\hat\theta^{(j)} - \Phi\theta_*\Big\|_2^2\Big] &= \frac{1}{n}\mathbb{E}_\varepsilon\Big[\big\|\mathbb{E}_{S^{(1)}}\big[\Phi\hat\theta^{(1)}\big] - \Phi\theta_*\big\|_2^2\Big] \\
&\qquad + \frac{1}{nm}\mathbb{E}_{\varepsilon,S^{(1)}}\Big[\big\|\Phi\hat\theta^{(1)} - \mathbb{E}_{S^{(1)}}\Phi\hat\theta^{(1)}\big\|^2\Big] \\
&= \frac{1}{n}\mathbb{E}_\varepsilon\Big[\big\|\Phi\hat\theta_{\mathrm{OLS}} - \Phi\theta_*\big\|_2^2\Big] + \sigma^2\frac{d}{nm}\frac{n-d}{s-d-1} \\
&= \sigma^2\frac{d}{n} + \sigma^2\frac{d}{nm}\frac{n-d}{s-d-1}.
\end{aligned}$$

Thus, when m or s tends to infinity, we recover the traditional OLS behavior, while for m and s finite, the performance degrades gracefully. Moreover, when $s = n$, even for $m = 1$, we get essentially twice the performance of the OLS estimator. We note that to get the same performance as OLS (up to a factor of 2), we need $m = \frac{n-d}{s-d-1} \sim \frac{n}{s}$ replications.

As in section 10.1, there is no statistical gain (here, compared to OLS), but only potentially a computational one (because some computations may be done in parallel and of reduced storage). See, for example, Dobriban and Liu (2019) for other criteria and sketching matrices.

Beyond Gaussian sketching. In this section, we have chosen a Gaussian sketching matrix S. This made the analysis simple because of the properties of the Gaussian distribution (invariance by rotation and availability of exact expectations for inverse Wishart distributions). The analysis can be extended with more complex tools to other random sketching matrices with more attractive computational properties, such as with many zeros, leading to subsampling observations or dimensions. See Wang et al. (2018), Dobriban and Liu (2019), and the references therein. For the random projections that follow, our analysis will apply to more general sketches.

[3] If $S \in \mathbb{R}^{a\times b}$ has independent standard Gaussian components, then $\mathbb{E}[(S^\top S)^{-1}] = \frac{1}{a-b-1} I$ if $a > b+1$, and $\mathbb{E}[SS^\top] = bI$; see https://en.wikipedia.org/wiki/Inverse-Wishart_distribution.

10.2.2 Random Projections

We also consider the fixed design setup, with a design matrix $\Phi \in \mathbb{R}^{n\times d}$ and a response vector of the form $y = \Phi\theta_* + \varepsilon$. We now assume that $d > n$ (high-dimensional setup) and the rank of Φ is n. In this high-dimensional setup, we need some form of regularization, which will come here from random projections.

For each $j \in \{1, \dots, n\}$, we consider a sketching matrix $S^{(j)} \in \mathbb{R}^{d\times s}$, for $s \leqslant n$ sampled independently from a distribution to be determined (we only assume that almost surely, its rank is equal to s). We then consider $\hat{\eta}^{(j)}$ as a minimizer of $\min_{\eta\in\mathbb{R}^s} \|y - \Phi S^{(j)}\eta\|_2^2$. For simplicity, we assume that matrix $\Phi S^{(j)}$ has rank s, which is the case almost surely for Gaussian projections; this implies that $\hat{\eta}^{(j)}$ is unique, but our result applies in all situations, as we are only interested in the denoised response vector. We now consider the average $\hat{\theta} = \frac{1}{m}\sum_{j=1}^m S^{(j)}\hat{\eta}^{(j)}$.

We thus consider the estimator $\hat{\eta}^{(j)} = \big((S^{(j)})^\top\Phi^\top\Phi S^{(j)}\big)^{-1}(S^{(j)})^\top\Phi^\top y \in \mathbb{R}^s$, obtained from the normal equation $(S^{(j)})^\top\Phi^\top\Phi S^{(j)}\hat{\eta}^{(j)} = (S^{(j)})^\top\Phi^\top y$ with the denoised response vector

$$\hat{y}^{(j)} = \Phi S^{(j)}\hat{\eta}^{(j)} = \Phi S^{(j)}\big((S^{(j)})^\top\Phi^\top\Phi S^{(j)}\big)^{-1}(S^{(j)})^\top\Phi^\top y \in \mathbb{R}^n.$$

Denoting $\Pi^{(j)} = \Phi S^{(j)}\big((S^{(j)})^\top\Phi^\top\Phi S^{(j)}\big)^{-1}(S^{(j)})^\top\Phi^\top$, it takes the form $\hat{y}^{(j)} = \Pi^{(j)}y$. Matrix $\Pi^{(j)}$ is almost surely an orthogonal projection matrix into an s-dimensional vector space, and its expectation is denoted as $\Delta \in \mathbb{R}^{n\times n}$, which satisfies $\mathrm{tr}(\Delta) = s$. We have, moreover, $0 \preccurlyeq \Delta \preccurlyeq I$; that is, all eigenvalues of Δ are between 0 and 1.

We can then compute expectations and variances as follows:

$$\begin{aligned}
\mathbb{E}_{S^{(j)}}\big[\hat{y}^{(j)}\big] &= \mathbb{E}_{S^{(j)}}\big[\Pi^{(j)}y\big] = \Delta y = \Delta\big[\Phi\theta_* + \varepsilon\big] = \Delta\varepsilon + \Delta\Phi\theta_* \\
\mathbb{E}_{S^{(j)}}\big[\hat{y}^{(j)}\big] - \Phi\theta_* &= \Delta\varepsilon + [\Delta - I]\Phi\theta_* \qquad (10.1)\\
\mathbb{E}_{S^{(j)}}\big[\big\|\hat{y}^{(j)} - \mathbb{E}_{S^{(j)}}\big[\hat{y}^{(j)}\big]\big\|_2^2\big] &= \mathbb{E}_{S^{(j)}}\big[\big\|(\Pi^{(j)} - \Delta)y\big\|_2^2\big] = y^\top\mathbb{E}_{S^{(j)}}\big[\big(\Pi^{(j)} - \Delta\big)^2\big]y \\
&= y^\top\mathbb{E}_{S^{(j)}}\big[\Pi^{(j)} - \Delta\Pi^{(j)} - \Pi^{(j)}\Delta + \Delta^2\big]y \ \text{ since } \Pi^{(j)}\Pi^{(j)} = \Pi^{(j)}, \\
&= y^\top\big(\Delta - \Delta^2\big)y, \ \text{since } \mathbb{E}[\Pi^{(j)} = \Delta. \qquad (10.2)
\end{aligned}$$

Thus, the overall (fixed design) expected generalization error is equal to, using that $S^{(1)}, \dots, S^{(m)}$ are i.i.d. matrices,

$$\begin{aligned}
&\frac{1}{n}\mathbb{E}_{\varepsilon,S}\Big[\big\|\tfrac{1}{m}\textstyle\sum_{j=1}^m \hat{y}^{(j)} - \Phi\theta_*\big\|_2^2\Big] \\
= \ &\frac{1}{n}\mathbb{E}_{\varepsilon}\Big[\big\|\mathbb{E}_{S^{(1)}}\big[\hat{y}^{(1)}\big] - \Phi\theta_*\big\|_2^2 + \frac{1}{m}\mathbb{E}_{S^{(1)}}\big[\big\|\hat{y}^{(1)} - \mathbb{E}_{S^{(1)}}\big[\hat{y}^{(1)}\big]\big\|_2^2\big]\Big] \\
&\qquad\qquad \text{by taking expectations with respect to all } S^{(j)}, \\
= \ &\frac{1}{n}\mathbb{E}_{\varepsilon}\Big[\big\|\Delta\varepsilon + [\Delta - I]\Phi\theta_*\big\|_2^2 + \frac{1}{m}y^\top\big(\Delta - \Delta^2\big)y\Big] \text{ using equations (10.1) and (10.2).}
\end{aligned}$$

Using the model $y = \Phi\theta_* + \varepsilon$ and the fact that $\mathbb{E}[\varepsilon] = 0$ and $\mathbb{E}[\varepsilon\varepsilon^\top] = \sigma^2 I$, we get

$$\begin{aligned}
& \frac{1}{n}\mathbb{E}_{\varepsilon,S}\Big[\big\|\tfrac{1}{m}\textstyle\sum_{j=1}^m \hat{y}^{(j)} - \Phi\theta_*\big\|_2^2\Big] \\
= \;& \frac{\sigma^2}{n}\operatorname{tr}(\Delta^2) + \frac{1}{n}\theta_*^\top\Phi^\top[I-\Delta]^2\Phi\theta_* + \frac{1}{nm}\big[\sigma^2(\operatorname{tr}(\Delta) - \operatorname{tr}(\Delta^2)) + \theta_*^\top\Phi^\top\big(\Delta - \Delta^2\big)\Phi\theta_*\big] \\
& \qquad \text{using the model } y = \Phi\theta_* + \varepsilon \text{ and the fact that } \mathbb{E}[\varepsilon] = 0 \text{ and } \mathbb{E}[\varepsilon\varepsilon^\top] = \sigma^2 I, \\
= \;& \frac{\sigma^2}{n}\big(1 - \frac{1}{m}\big)\operatorname{tr}(\Delta^2) + \frac{\sigma^2 s}{nm} + \frac{1}{n}\theta_*^\top\Phi^\top[\Delta - I]^2\Phi\theta_* + \frac{1}{nm}\theta_*^\top\Phi^\top\big(\Delta - \Delta^2\big)\Phi\theta_* \\
= \;& \frac{\sigma^2}{n}\big(1 - \frac{1}{m}\big)\operatorname{tr}(\Delta^2) + \frac{\sigma^2 s}{nm} + \frac{1}{n}\theta_*^\top\Phi^\top\big[I - \Delta + \big(\frac{1}{m} - 1\big)(\Delta - \Delta^2)\big]\Phi\theta_* \\
\leqslant \;& \frac{\sigma^2 s}{n} + \frac{1}{n}\theta_*^\top\Phi^\top\big[I - \Delta\big]\Phi\theta_*, \text{ since } \Delta^2 \preccurlyeq \Delta, \qquad (10.3)
\end{aligned}$$

which is the value for $m = 1$ (single replication). Note that the expectation (before taking the bound) decreases in m, with a limit $\frac{\sigma^2\operatorname{tr}(\Delta^2)}{n} + \frac{1}{n}\theta_*^\top\Phi^\top[I-\Delta]^2\Phi\theta_*$ when $m \to +\infty$ (with improved bias and variance terms). We now follow Kabán (2014) and Thanei et al. (2017) to bound the matrix $I - \Delta$.

Since Δ is the expectation of a projection matrix, we already know that $0 \preccurlyeq \Delta \preccurlyeq I$. We omit the superscript $^{(j)}$ for clarity, and we consider $\Pi = \Phi S\big(S^\top\Phi^\top\Phi S\big)^{-1}S^\top\Phi$. For any vector $z \in \mathbb{R}^n$, we consider

$$\begin{aligned}
z^\top(I-\Delta)z \;&=\; \mathbb{E}_S\big[z^\top(I-\Pi)z\big] = \mathbb{E}_S\big[z^\top z - z^\top\Phi S\big(S^\top\Phi^\top\Phi S\big)^{-1}S^\top\Phi^\top z\big] \\
&=\; \mathbb{E}_S\Big[\min_{u\in\mathbb{R}^s}\|z - \Phi S u\|_2^2\Big] \text{ by definition of projections,} \\
&\leqslant\; \mathbb{E}_S\Big[\min_{v\in\mathbb{R}^d}\|z - \Phi S S^\top v\|_2^2\Big] \text{ by minimizing over a smaller subspace,} \\
&\leqslant\; \min_{v\in\mathbb{R}^d}\mathbb{E}_S\Big[\|z - \Phi S S^\top v\|_2^2\Big] \text{ by properties of the expectation.}
\end{aligned}$$

We can expand this to get

$$\mathbb{E}_S\Big[\|z - \Phi S S^\top v\|_2^2\Big] \;=\; \|z\|_2^2 - 2z^\top\Phi\mathbb{E}_S\big[SS^\top\big]v + v^\top\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]v,$$

leading to, after selecting the optimal v as $v = \big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top z$,

$$z^\top(I-\Delta)z \;\leqslant\; z^\top\Big(I - \Phi\mathbb{E}_S\big[SS^\top\big]\big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top\Big)z.$$

We then need to apply to $z = \Phi\theta_*$ and get

$$\theta_*^\top\Phi^\top\big[I-\Delta\big]\Phi\theta_* \leqslant \theta_*^\top\Phi^\top\Big(I - \Phi\mathbb{E}_S\big[SS^\top\big]\big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top\Big)\Phi\theta_*.$$

Thus, we get an overall upper bound of

$$\frac{\sigma^2 s}{n} + \frac{1}{n}\theta_*^\top\Phi^\top\Big(I - \Phi\mathbb{E}_S\big[SS^\top\big]\big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top\Big)\Phi\theta_*,$$

composed of expectations that can be readily computed. As shown next for special cases, we obtain a bias-variance trade-off similar to equation (3.6) for ridge regression in section 3.6, but now with random projections. Note that in the fixed design setting, there is no explosion of the testing error when $s = n$, as opposed to the random design setting studied in section 12.2 in the context of "double descent" (where generalization to unseen inputs is required).

Gaussian projections. If we assume Gaussian random projections, with $S \in \mathbb{R}^{d\times s}$ with independent standard Gaussian components, we get, from properties of the Wishart distribution,[4]

$$\mathbb{E}_S[SS^\top] = sI \text{ and } \mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big] = s(s+1)\Phi^\top\Phi + s\operatorname{tr}(\Phi^\top\Phi)I.$$

We then get

$$\begin{aligned}
\theta_*^\top\Phi^\top\big[I-\Delta\big]\Phi\theta_* &\leqslant \theta_*^\top\Phi^\top\Big(I - \Phi\mathbb{E}_S\big[SS^\top\big]\big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top\Big)\Phi\theta_* \\
&= \theta_*^\top\Phi^\top\Big(I - s^2\Phi\big(s(s+1)\Phi^\top\Phi + s\operatorname{tr}(\Phi^\top\Phi)I\big)^{-1}\Phi^\top\Big)\Phi\theta_* \\
&= \theta_*^\top\Phi^\top\Phi\theta_* - s\theta_*(\Phi^\top\Phi)^2\big((s+1)\Phi^\top\Phi + \operatorname{tr}(\Phi^\top\Phi)I\big)^{-1}\theta_* \\
&= \theta_*^\top\Phi^\top\Phi\big(\Phi^\top\Phi + \operatorname{tr}(\Phi^\top\Phi)I\big)\big((s+1)\Phi^\top\Phi + \operatorname{tr}(\Phi^\top\Phi)I\big)^{-1}\theta_* \\
&\leqslant 2\operatorname{tr}(\Phi^\top\Phi)\cdot\theta_*^\top\Phi^\top\Phi\big((s+1)\Phi^\top\Phi + \operatorname{tr}(\Phi^\top\Phi)I\big)^{-1}\theta_* \\
&\quad \text{using that } \Phi^\top\Phi + \operatorname{tr}(\Phi^\top\Phi)I \text{ has eigenvalues less than } 2\operatorname{tr}(\Phi^\top\Phi), \\
&\leqslant 2\operatorname{tr}(\Phi^\top\Phi)\frac{\|\theta_*\|_2^2}{s+1},
\end{aligned}$$

since $\Phi^\top\Phi\big((s+1)\Phi^\top\Phi + \operatorname{tr}(\Phi^\top\Phi)I\big)^{-1}$ has eigenvalues less than $1/(s+1)$. The overall excess risk is then less than

$$\frac{\sigma^2 s}{n} + \frac{2}{n}\operatorname{tr}(\Phi^\top\Phi)\frac{\|\theta_*\|_2^2}{s+1}, \tag{10.4}$$

which is exactly of the form obtained for ridge regression in equation (3.6) with the identification $s \sim \frac{\operatorname{tr}(\Phi^\top\Phi)}{\lambda}$. We can consider other sketching matrices with additional properties, such as sparsity (see exercise 10.5).

Exercise 10.5 *We consider a sketching matrix $S \in \mathbb{R}^{d\times s}$, where each column is equal to one of the d canonical basis vectors of $\mathbb{R}^d$, selected uniformly at random and independently. Compute $\mathbb{E}[SS^\top]$, as well as $\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]$, as well as a bound similar to equation (10.4).*

[4] If $W = S_1S_1^\top$ for $S_1 \in \mathbb{R}^{n\times s}$ with independent standard Gaussian components, then $\mathbb{E}[W] = sI$ and for an $n\times n$ diagonal matrix D, we have $\mathbb{E}[WD^2W] = s(s+1)D^2 + s\operatorname{tr}(D^2)I$.

Kernel methods. (♦) The random projection idea can be extended to kernel methods discussed in chapter 7. We consider the kernel matrix $K = \Phi\Phi^\top \in \mathbb{R}^{n\times n}$, and the assumption $y = \Phi\theta_* + \varepsilon$ with $\|\theta_*\|_2$ bounded is turned into $y = y_* + \varepsilon$ with $y_*^\top K^{-1} y_*$ bounded. This corresponds to $y_* = K\alpha$, with a reproducing kernel Hilbert space (RKHS) norm $\alpha^\top K\alpha$. We then consider a random sketch $\hat\Phi \in \mathbb{R}^{n\times s}$ and an approximate kernel matrix $\hat K$. We then obtain an estimate $\hat y = \hat\Phi(\hat\Phi^\top\hat\Phi)^{-1}\hat\Phi^\top y$. Matrix Π is then $\Pi = \hat\Phi(\hat\Phi^\top\hat\Phi)^{-1}\hat\Phi^\top$, and for the analysis, we need to compute its expectation, Δ (this corresponds to replacing ΦS in earlier developments starting at the beginning of section 10.2.2 with $\hat\Phi$). We have, following the same reasoning as before, for an arbitrary deterministic $z \in \mathbb{R}^n$,

$$\begin{aligned}
z^\top(I-\Delta)z &= \mathbb{E}_{\hat\Phi}\big[z^\top(I-\Pi)z\big] = \mathbb{E}_{\hat\Phi}\big[z^\top z - z^\top\hat\Phi\big(\hat\Phi^\top\hat\Phi\big)^{-1}\hat\Phi^\top z\big] \\
&= \mathbb{E}_{\hat\Phi}\Big[\min_{u\in\mathbb{R}^s}\|z-\hat\Phi u\|_2^2\Big] \text{ by definition of projections,} \\
&\leqslant \mathbb{E}_{\hat\Phi}\Big[\min_{v\in\mathbb{R}^n}\|z-\hat\Phi\hat\Phi^\top v\|_2^2\Big] \text{ by minimizing over a smaller subspace,} \\
&\leqslant \min_{v\in\mathbb{R}^n}\mathbb{E}_{\hat\Phi}\Big[\|z-\hat\Phi\hat\Phi^\top v\|_2^2\Big] \text{ by properties of the expectation.}
\end{aligned}$$

We can expand this to get

$$\mathbb{E}_{\hat\Phi}\Big[\|z-\hat\Phi\hat\Phi^\top v\|_2^2\Big] = \|z\|_2^2 - 2z^\top\mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\big]v + v^\top\mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\hat\Phi\hat\Phi^\top\big]v,$$

leading to, after selecting the optimal v as $v = \big(\mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\hat\Phi\hat\Phi^\top\big]\big)^{-1}\mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\big]z$,

$$z^\top(I-\Delta)z \leqslant z^\top\Big(I - \mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\big]\big(\mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\hat\Phi\hat\Phi^\top\big]\big)^{-1}\mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\big]\Big)z. \tag{10.5}$$

We then need to apply equation (10.5) to $z = y_*$ to get

$$\theta_*^\top\Phi^\top\big[I-\Delta\big]\Phi\theta_* \leqslant y_*^\top\Big(I - \mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\big]\big(\mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\hat\Phi\hat\Phi^\top\big]\big)^{-1}\mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\big]\Big)y_*.$$

We can, for example, consider each column of $\hat\Phi$ to be sampled from a Gaussian distribution with mean zero and covariance matrix K, for which we have

$$\mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\big] = sK \quad\text{and}\quad \mathbb{E}_{\hat\Phi}\big[\hat\Phi\hat\Phi^\top\hat\Phi\hat\Phi^\top\big] = s(s+1)K^2 + s\operatorname{tr}(K)\cdot K.$$

Using equation (10.3), with the same derivations that led to equation (10.4), this leads to a bound on the expected excess risk equal to $\frac{\sigma^2 s}{n} + \frac{2}{n}\operatorname{tr}(K)\frac{y_*^\top K^{-1}y_*}{s+1}$, which is exactly the bound in equation (10.4) in the kernel context. However, it is not interesting in practice, as it requires the computation of the kernel matrix K and typically a square root to sample from the multivariate Gaussian distribution, which has a running-time complexity of $O(n^3)$.

In practice, many kernels come with a random feature expansion of the form $k(x,x') = \mathbb{E}_v\big[\varphi(x,v)\varphi(x',v)\big]$, such that $|\varphi(x,v)| \leqslant R$ almost surely (as presented in section 7.4).

We can take for each column of $\hat{\Phi}$ the vector $(\varphi(x_1, v), \dots, \varphi(x_n, v))^\top \in \mathbb{R}^n$ for a random independent v. Then we have $\mathbb{E}\big[\hat{\Phi}\hat{\Phi}^\top\big] = sK$ by construction, while a short calculation (left as an exercise) shows that the second-order moment can be bounded as

$$\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big] \preccurlyeq s(s-1)K^2 + sR^2K.$$

This leads to the bound $\frac{\sigma^2 s}{n} + \frac{1}{n}R^2\frac{y_*^\top K^{-1} y_*}{s-1}$, which is almost the same as before, but with an efficient practical algorithm (since we now have to solve a least-squares regression problem in dimension s, which is more efficient than using the kernel trick if $s < n$).

Experiments. In figure 10.2, we consider a polynomial regression problem in dimension $d_{\mathcal{X}} = 20$, with polynomials of a maximum degree of 2, and thus a feature space of dimension $d = 1 + d_{\mathcal{X}} + d_{\mathcal{X}}(d_{\mathcal{X}}+1)/2 = 231$. We also compare ridge regression with Gaussian random projections. We see better performance as m grows, consistent with our bounds (underfitting for small s, overfitting with large s). Moreover, when the number m of times the dataset is randomly projected goes from 10 to 100, we obtain almost the same plot, with better performance than $m = 1$, highlighting the fact that $m = 1$ is not optimal but taking m too large is not useful.

Johnson-Lindenstrauss lemma (♦). A related classical result in Gaussian random projections shows that n feature vectors $\varphi_1, \dots, \varphi_n \in \mathbb{R}^d$ can be well represented in dimension s by Gaussian random projections, with s growing only logarithmically in n, and independent of the underlying dimension. Lemma 10.1 shows that all pairwise distances are preserved (a small modification would lead to the preservation of all dot products).

Lemma 10.1 (Johnson and Lindenstrauss, 1984) *Given $\varphi_1, \dots, \varphi_n \in \mathbb{R}^d$, let $S \in \mathbb{R}^{d\times s}$ be a random matrix with independent standard Gaussian random variables. Then, for any $\varepsilon \in (0, 1/2)$ and $\delta \in (0, 1)$, if $s \geqslant \frac{6}{\varepsilon^2}\log\frac{n^2}{\delta}$, with probability greater than $1-\delta$, we have*

$$\forall i, j \in \{1, \dots, n\},\ (1-\varepsilon)\|\varphi_i - \varphi_j\|_2^2 \leqslant \big\|s^{-1/2}S^\top\varphi_i - s^{-1/2}S^\top\varphi_j\|_2^2 \leqslant (1+\varepsilon)\|\varphi_i - \varphi_j\|_2^2. \tag{10.6}$$

Proof (♦) Let $\psi \in \mathbb{R}^d$ with the ℓ_2-norm equal to 1. The random variable $y = \psi^\top SS^\top\psi$ is the sum of s random variables $\psi^\top S_{\cdot j}S_{\cdot j}^\top\psi$, for $S_{\cdot j}$ the jth column of S, $j \in \{1, \dots, s\}$. Each of these is the square of $S_{\cdot j}^\top\psi$, which is Gaussian with mean zero and variance equal to $\|\psi\|_2^2 = 1$. Thus, y is a chi-squared random variable.[5] We can thus apply concentration results from exercise 8.1, leading to

$$\mathbb{P}\big(|y - s| \geqslant s\varepsilon\big) \leqslant \Big(\frac{1-\varepsilon}{\exp(-\varepsilon)}\Big)^{s/2} + \Big(\frac{1+\varepsilon}{\exp(\varepsilon)}\Big)^{s/2}.$$

We can then use the inequality $\log(1+u) \leqslant u - \frac{u^2}{3}$ for any $|u| \leqslant \frac{1}{2}$, applied to ε and $-\varepsilon$, leading to the probability bound $\mathbb{P}\big(|y - s| \geqslant s\varepsilon\big) \leqslant 2\exp\big(-\frac{s}{2}\frac{\varepsilon^2}{3}\big)$. We then apply

[5] See https://en.wikipedia.org/wiki/Chi-squared_distribution.

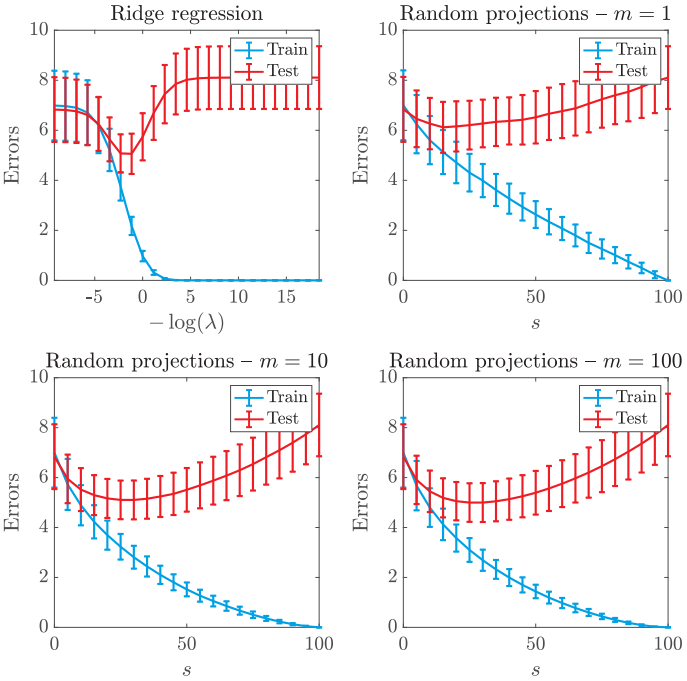

Figure 10.2. Polynomial regression in dimension 20, with polynomials of a maximum degree of 2, with $n = 100$. Top left: training and testing errors for ridge regression in the fixed design setting (the input data are fixed, and only the noise variables are resampled for computing the test error). All other plots: training and testing errors for Gaussian random projections, with different numbers of random projections: $m = 1$ (top right), $m = 10$ (bottom left), and $m = 100$ (bottom right). All the curves are averaged over 100 replications of the noise variables and the random projections.

this bound for ψ being the $n(n-1)/2$ vectors $\varphi_i - \varphi_j$, for $i \neq j$, leading to, using a union bound, a probability that equation (10.6) is not satisfied with a probability less than $n^2 \exp(-s\varepsilon^2/6)$, leading to the desired result. ■

In our context of least-squares regression, the Johnson-Lindenstrauss lemma shows that the kernel matrix is preserved by random projections so that predictions with the projected data should be close to predictions with the original data. The results in this section provide a direct proof that aims to characterize directly the predictive performance of such random projections (using the Johnson-Lindenstrauss lemma to obtain similar bounds is not straightforward as we consider unregularized regression, where perturbations of matrix inverses are harder to control).

10.3 Boosting

In sections 10.1 and 10.2, we focused on uniformly combining the outputs (e.g., plain averaging) of estimators obtained by randomly reweighted versions of the original datasets. Reweighting was performed independent of the performance of the resulting prediction functions, and the training procedures for all predictors could be done in parallel. In this section, we explore *sequential* reweightings of the training datasets that depend on the mistakes made by the current prediction functions. While the natural parallelizability is lost, we will see that we get additional statistical benefits.

In the early boosting procedures adapted to binary classification, the original learning algorithms (going from datasets to prediction functions with binary values) were used directly on a reweighted version, such as Adaboost (see, e.g., Freund et al., 1999). Our analysis will be carried out for boosting procedures, often referred to as "gradient boosting," which are adapted to real-valued outputs, as done in the rest of this book (noting that for classification, we can use convex surrogates).

The theory of boosting is rich, with many connections, and in this section, we only provide a consistency proof in the simplest setting. See Schapire and Freund (2012) for more details.

10.3.1 Problem Setup

Given an input space $\mathcal{X}$ and n observations $(x_i, y_i) \in \mathcal{X} \times \mathbb{R}$, $i = 1, \dots, n$, we are given a set of predictors $\varphi(\cdot, w) : \mathcal{X} \to \mathbb{R}$, for $w \in \mathcal{W}$, with $\mathcal{W}$ typically being a compact subset of a finite-dimensional vector space.

The main assumption is that given weights $\alpha \in \mathbb{R}^n$, one can reasonably easily find the function $\varphi(\cdot, w)$ that minimizes with respect to $w \in \mathcal{W}$:

$$\sum_{i=1}^{n} \alpha_i \varphi(x_i, w); \tag{10.7}$$

that is, the dot product between α and the n outputs of $\varphi(\cdot, w)$ on the n observations. In this section, for simplicity, we assume that this minimization can be done exactly. This is often referred to as the "weak learner" assumption. Many examples are available, such as the following:

- Linear stumps for $\mathcal{X} = \mathbb{R}^d$: $\varphi(x, w) = \pm(w_0^\top x + w_1)_+$, where $w = (w_0, w_1) \in \mathbb{R}^d \times \mathbb{R}$, with sometimes the restriction that w has a single nonzero component (where the weak learning tractability assumption is indeed verified; see exercise 10.6). This will lead to a predictor, which is a one-hidden-layer neural network as presented in chapter 9, but learned sequentially rather than by GD on the empirical risk. In the context of binary classification, the weak learners are sometimes thresholded to values in $\{-1, 1\}$ by taking their signs.

  Exercise 10.6 *For linear stumps with only one nonzero coordinate for the slope, show how to minimize equation (10.7) efficiently.*