IST Austria: Statistical Machine Learning 2020/21
Christoph Lampert <chl@ist.ac.at>
TAs: Alexandra Peste <alexandra.peste@ist.ac.at>,
Bernd Prach <bernd.prach@ist.ac.at>
Exercise Sheet 4/5 (due date 02/11/2020, 10:15am)
Please send your solutions via email to the TAs

1 Weighted Estimators

Let $w_1, \dots, w_n$ be non-negative constants with $\frac{1}{n}\sum_i w_i = 1$. For a training set $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\} \overset{i.i.d.}{\sim} p$ and a prediction function $f : \mathcal{X} \to \mathcal{Y}$, define a *weighted estimator*

$$\hat{\mathcal{R}}^w(f) := \frac{1}{n}\sum_{i=1}^{n} w_i \ell(y_i, f(x_i)) \tag{1}$$

a) Show that $\hat{\mathcal{R}}^w(f)$ is an unbiased estimator of $\mathcal{R}(f)$.

b) Show that its variance is $\frac{\sigma^2}{n^2}\|w\|^2$, where $\sigma^2 = \mathrm{Var}[\ell(y, f(x))]$ is the variance of a single loss term

note: red indicates a correction of the original sheets, blue is a additional explanation

c) Derive by constrained optimization: which choice of w yields the estimator of smallest variance?

d) Derive by constrained optimization: which choices of w yield the estimator of largest variance?

e) In Lecture 6, we see several weighted estimators that use samples from a distribution p_S to estimate the risk for a (different) distribution p_T. How is this not a contradiction to a)?

2 Model selection can fail

Cross-validation and leave-one-out estimation (Lecture 5) are popular techniques for selecting hyperparameters of a learning system. Show that they are not guaranteed to work, though, by constructing a learning situation (in particular a data distribution, loss function, hypothesis class, etc), in which the leave-one-out error of a hypothesis is a terrible approximation to its generalization error, regardless of training set size.

3 Excurse: Ridge Regression

Regularized linear least-squares regression is also called *ridge regression (RR)*. For a regularization constant, $\lambda > 0$, and a training set, $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R}$, it learns a predictor $h(x) = w_{\mathrm{RR}}^\top x$ by solving

$$\min_{w \in \mathbb{R}^d} \quad \lambda\|w\|^2 + \sum_{i=1}^{n} (w^\top x_i - y_i)^2 \tag{2}$$

a) Show that ridge regression has the closed-form solution

$$w_{\mathrm{RR}} = (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I}_{d\times d})^{-1}\mathbf{X}\mathbf{y}$$

where $\mathbf{X} = (x_1|\dots|x_n) \in \mathbb{R}^{d\times n}$ is the matrix that contains the training data vectors as columns, $\mathbf{y} = (y_1, \dots, y_n) \in \mathbb{R}^n$ is the vector of target values, and $\mathbf{I}_{d\times d}$ is the $(d \times d)$-identity matrix.

b) Show that an alternative way for computing w_{RR} is

$$w_{\mathrm{RR}} = \mathbf{X}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{n\times n})^{-1}\mathbf{y} \tag{3}$$

c) In practice, when would you prefer to use (3) over (2), or vice versa?

Hint: if you're stuck on b), use the matrix identity $(P^{-1} + B^\top R^{-1} B)^{-1} B^\top R^{-1} = P B^\top (B P B^\top + R)^{-1}$ for any P, B, R where it makes sense (i.e. P and R invertible, B not necessarily square but of matching dimensions). Can you prove the identity?

4 Efficient leave-one-out error estimation

As discussed in the lecture, computing the leave-one-out (LOO) error is in general computationally very costly. In this exercise we show that this is not the case for ridge regression, where leave-one-out error estimation can be performed efficiently without repeated training.
For this exercise, you can make use of the results from Exercise 3, even if you did not prove them.

- Let $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R}$ be a training set. For any $i = 1, \ldots, n$, denote by $\mathcal{D}_i$ the set of size $m - 1$ in which (x_i, y_i) was removed from $\mathcal{D}$.

- For any dataset S, denote by h_S the predictor obtained by training ridge regression on S (see Exercise 3).

- By definition, the leave-one-out error for squared loss, $\ell(y, y') = (y - y')^2$, with respect to $\mathcal{D}$ is

$$\hat{\mathcal{R}}_{\mathrm{LOO}} = \frac{1}{n}\sum_{i=1}^{n} (h_{\mathcal{D}_i}(x_i) - y_i)^2 \tag{4}$$

a) Let $\mathcal{D}_i'$ be the set $\mathcal{D}$ with the value y_i replaced by $h_{\mathcal{D}_i}(x_i)$. Show that $h_{\mathcal{D}_i'} = h_{\mathcal{D}_i}$.

b) Denote by $\mathbf{y}_i$ the vector of target values, $\mathbf{y}$, with the i-th entry replaced by $h_{\mathcal{D}_i}(x_i)$, i.e.
$\mathbf{y}_i = \mathbf{y} - y_i \mathbf{e}_i + h_{S_i}(x_i)\mathbf{e}_i$, where $\mathbf{e}_i = (0, \ldots, 0, \overbrace{1}^{i\text{-th pos}}, 0, \ldots, 0) \in \mathbb{R}^n$. Show that

$$h_{\mathcal{D}_i}(x_i) = \mathbf{y}_i^\top (\mathbf{K} + \lambda \mathbf{I}_{n\times n})^{-1}\mathbf{K}\mathbf{e}_i \qquad \text{for } \mathbf{K} = \mathbf{X}^\top \mathbf{X}.$$

c) Show that the leave-one-out error for $h_{\mathcal{D}}$ can be written in the following way

$$\hat{\mathcal{R}}_{\mathrm{LOO}} = \frac{1}{n}\sum_{i=1}^{n} \Big[\frac{h_{\mathcal{D}}(x_i) - y_i}{\mathbf{e}_i(\mathbf{K} + \lambda \mathbf{I}_{n\times n})^{-1}\mathbf{K}\mathbf{e}_i}\Big]^2 \tag{5}$$

Note that (5) can be evaluated efficiently, because the values showing up in the denominator of (5) are simply the diagonal entries of $\mathbf{M} = (\mathbf{K} + \lambda \mathbf{I}_{n\times n})^{-1}\mathbf{K}$, so only a single $(n \times n)$-matrix inversion is needed to compute all of them.

d) Suppose that all diagonal entries of $\mathbf{M}$ are identical with value γ. Then how does the leave-one-out error $\hat{\mathcal{R}}_{\mathrm{LOO}}$ relate to the training error $\hat{\mathcal{R}}_{\mathcal{D}} = \frac{1}{n}\sum_{i=1}^{n}(h_{\mathcal{D}}(x_i) - y_i)^2$? Is there a value for which they coindice? If yes, could this happen in practice?

5 Practical Experiments IV

Expand your system from the *Practical Experiments III*.

a) Model selection

- perform automatic model selection to select the regularization strength, λ. Either use $\mathcal{D}_{\text{eval}}$ as a validation set, or apply one of the alternative methods, such as 5-fold cross-validation. If you chose the latter, please do not use an existing library function but implement data splitting and the repeated calls to classifier training yourself.
- retrain your system with the regularization parameter that you've obtained to obtain a single final predictor.
- afterwards (and only then), download the test data *XtestIMG.txt* and *Ytest.txt* from the homepage and compute the test error of the single final predictor.

b) Domain adaptation

- download the data *XtrainIMG2.txt* and *Ytrain2.txt* from the homepage. These also contain images of eyes, but of *right eyes*, where the original data contained *left eyes*.
- rerun the pipeline from a)
  - using the right eyes data as input,
  - using the right eyes and left data combined data as input,
  - using the right eyes and left data combined data as input, where you first mirrored the right eyes horizontally to look like left ones

  In all cases, evaluate the final predictors (and only these) on the original test data.