Exercise 6

Deadline: 13.12.2017, 10:00

This exercise is dedicated to regularized regression.

Regulations

Please hand in your solution as a Jupyter notebook `regularized-regression.ipynb`, accompanied with exported PDF `regularized-regression.pdf`. Zip all files into a single archive with naming convention (sorted alphabetically by last names)

`lastname1-firstname1_lastname2-firstname2_exercise06.zip`

or (if you work in a team of three)

`lastname1-firstname1_lastname2-firstname2_lastname3-firstname3_exercise06.zip`

and upload it to Moodle before the given deadline. We will give zero points if your Zip file does not conform to the naming convention.

1 Bias and variance of ridge regression (8 points)

Ridge regression solves the regularized least squares problem

$$\widehat{\beta}_\tau = \text{argmin}_\beta \, \|X\beta - y\|_\text{F}^2 + \tau \, \|\beta\|_2^2$$

with regularization parameter $\tau \geq 0$. Regularization introduces some bias into the solution in order to achieve a potentially large gain in variance. Assume that the true model is $y = X\beta^* + \epsilon$ with zero-mean Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Prove that expectation and covariance matrix of the regularized solution (both taken over all training sets of size N) are then given by

$$\mathbb{E}[\widehat{\beta}_\tau] = S_\tau^{-1} S \; \beta^* \qquad \text{Cov}[\widehat{\beta}_\tau] = S_\tau^{-2} S \; \sigma^2$$

where S and S_τ are the ordinary and regularized scatter matrices:

$$S = X^\top X \qquad S_\tau = X^\top X + \tau \mathbb{1}$$

Notice that expectation and covariance reduce to the corresponding expressions of ordinary least squares (as derived in the lecture) when $\tau = 0$:

$$\mathbb{E}[\widehat{\beta}_{\tau=0}] = \beta^* \qquad \text{Cov}[\widehat{\beta}_{\tau=0}] = S^{-1} \; \sigma^2$$

Since S_τ is greater than S (in any norm), regularization has a shrinking effect on both expectation and covariance.

2 Denoising of a CT image (11 points)

In task 3 of exercise 5, we diminished the radiation dose for our patient by using fewer projection angles. However, this introduced noise in the resulting tomogram. Ridge regression offers a simple possibility to reduce the noise level[1].

[1] You will notice that the improvements don't look too impressive. Humans tend to prefer the noisy, but unbiased version of the image, because the eye's built-in noise suppression is way more powerful than ridge regression. In other words, the eye and the algorithm have different bias-variance trade-offs.

To apply regularization to tomographic reconstruction, the $D \times D$ diagonal matrix $\sqrt{\tau}\mathbb{1}$ must be appended at the bottom of the weight matrix X from exercise 5:

$$X' = \begin{bmatrix} X \\ \sqrt{\tau}\mathbb{1} \end{bmatrix}$$

Correspondingly, vector y must be extended with D zeros.

Add a new parameter to the function `construct_X()` to activate regularization

`X = construct_X(M, alphas, Np = None, tau=0)`

For `tau=0`, the function shall behave exactly like the original version from exercise 5. If `tau>0`, it shall return the augmented matrix X'. Reconstruct the tomogram for 64 angles with $\tau = 0, 1, 10, 100, 1000, 10000$ and display the resulting CT images. Find the value of τ with the best compromise of image sharpness and noise.

Compare the ridge regression results with Gaussian filtering, another popular noise reduction technique, which replaces each pixel with a weighted average of its neighbors (see `https://en.wikipedia.org/wiki/Gaussian_filter` for more details). A suitable implementation is provided by function `gaussian_filter` in module `scipy.ndimage.filters`. Apply `gaussian_filter` with filter sizes `sigma = 1,2,3,5,7` to the CT image with $\tau = 0$, display the results and comment on the similarities and differences between ridge regression and Gaussian filtering.

3 Automatic feature selection for regression

Sparse regression regularizes the solution such that unimportant elements of $\widehat{\beta}$ are set to zero. The corresponding columns of X could be dropped from the matrix without any effect on the solution – the remaining columns are obviously more useful as an explanation for the response y. Therefore, sparse regression can be interpreted as a method for *relevant* feature identification.

In exercise 2, you implemented dimension reduction from a full image of a digit to just two features by selecting important pixels manually. In this exercise, we will automatically find relevant pixels by means of sparse regression.

3.1 Implement Orthogonal Matching Pursuit (5 points)

"Orthogonal Matching Pursuit"[2] is a simple greedy sparse regression algorithm. It approximates the exact algorithms for least squares with L_0 or L_1 regularization (cf. the lecture for details) and is defined as:

Initialization:

- Inputs: $X \in \mathbb{R}^{N \times D}$, $y \in \mathbb{R}^N$, $T > 0$ (the desired number of non-zero elements in the final solution $\widehat{\beta}^{(T)}$)
- Define the initial sets of active resp. inactive columns as $A^{(0)} = \emptyset$ and $B^{(0)} = \{j : j = 1 \dots D\}$.
- Define the initial residual $r^{(0)} = y$.

Iteration: for $t = 1 \dots T$ do

1. Find the inactive column that has maximal correlation with the current residual:

$$j^{(t)} = \text{argmax}_{j \in B} \left| X_j^\top \cdot r^{(t-1)} \right|$$

2. Move $j^{(t)}$ from the inactive set B to the active set A.

[2] not to be confused with its simpler cousin "Matching Pursuit"

3. Form the active matrix $X^{(t)}$ consisting of all currently active columns of X.
4. Solve the least squares problem

$$\widehat{\beta}^{(t)} = \text{argmin}_{\beta} \left\| X^{(t)} \beta - y \right\|_{\text{F}}^2$$

5. Update the residual $r^{(t)} = y - X^{(t)} \widehat{\beta}^{(t)}$.

Implement this algorithm as a function

```
solutions = omp_regression(X, y, T)
```

which returns $\widehat{\beta}^{(t)}$ for $t = 1 \ldots T$ as a $D \times T$ matrix, i.e. the inactive elements in each solution are not dropped, but explicitly set to zero.

3.2 Classification with sparse LDA (8 points)

We again use the `digits` dataset of scikit-learn and classify '1' vs. '7'. For balanced training sets, LDA can be formulated as a least squares problem, where the rows X_i are the flattened images of each digit, and

$$y_i = \begin{cases} 1 & \text{if instance } i \text{ is a '1'} \\ -1 & \text{if instance } i \text{ is a '7'} \end{cases}$$

are the desired responses. Now execute `omp_regression()` with sufficiently big T to get the sequence of sparse LDA solutions for $t = 1 \ldots T$.

Report the error rate on the test set for $t = 1 \ldots T$. How many pixels should be used for acceptable error rates? Is it necessary/beneficial to standardize the data before training and testing?

Visualize in which order the pixels are switched to *active* as t increases, and show if a pixel votes in favour or against class '1'. What is a good criterion for this distinction? Compare these results with your hand-crafted feature selection in exercise 2 – did you select the same pixels?

3.3 One-against-the-rest classification (8 points)

A *one-against-the-rest classifier* is actually a collection of C classifiers, one for each class k. To construct it, one uses the given training data to create C auxiliary training sets with target responses

$$y_i^{(k)} = \begin{cases} 1 & \text{if instance } i \text{ has class } k \\ -1 & \text{if instance } i \text{ has class } \neq k \end{cases}$$

The positive subset contains all instances of class k, and the negative subset an equally sized random selection from all other classes. Use these auxiliary training sets to train a sparse LDA for the digits $k = 0 \ldots 9$ with suitable T (all classifiers should use the same T).

At testing, a new instance is rated by all C classifiers and assigned to the class whose classifier returns the largest response, or to the additional class "unknown" if all classifiers have negative response. Report the confusion matrix of your predictor on the test set.

Why is it useful to introduce class "unknown"? Parameterize your classifier such that a few test instances are actually assigned to this class and visualize these instances. What do you observe?