Computer Lab: Quantile Regression

Convex analysis, monotone operators and optimization
Olivier Fercoq olivier.fercoq@telecom-paristech.fr
14 December 2017

This computer lab is in three parts.

1. Before 14 December, you should download the data, check that you are able to run the attached code using Spark and answer a few preliminary questions. This first part is to be done at home. You may work in groups as large as you wish and share your results between groups.

2. On 14 December 13:00-16:15, we will have the actual computer lab. We booked room C130 of Telecom ParisTech. For this second part, you will be in groups of 1 or 2 persons.

3. Before 17 December, you should:

   - send to `olivier.fercoq@telecom-paristech.fr` your code in a ipython notebook
   - send a pdf file with the answers to the theoretical questions.

1 Data

We will be using the Gas sensor ethylene_methane dataset for this computer lab. Please download the dataset on
`https://archive.ics.uci.edu/ml/datasets/Gas+sensor+array+under+dynamic+gas+mixtures`
There are two files. We will only work on `ethylene_methane.txt`.

We will consider the first half of the observations as training set and the second half as testing set. We shall discard column 0 which is only a time stamp. We will consider columns 1 and 2 as observations. These are the gas concentrations. The features are columns 3 to 18.

As the dataset is large, we would like to compute the estimators using parallel computations. We will base on Spark to perform the parallelization. When developing the algorithms, it may be useful to firstly use only a subset of the training set.

2 Spark

1. If you are using a computer of Télécom ParisTech, check that you can access `/cal/softs/spark/`. Otherwise, install Spark from `http://spark.apache.org/downloads.html`.

2. To launch a jupyter notebook that allows pyspark commands and uses 4 processors, type

   `PYSPARK_DRIVER_PYTHON_OPTS=notebook PYSPARK_DRIVER_PYTHON=ipython MASTER=local[4] path/to/spark/bin/pyspark`

3. Read the Spark programming guide on

   `http://spark.apache.org/docs/latest/quick-start.html`

Question 2.1
Comment the Preparation section of the joint Jupyter notebook. Explain in particular what the Spark functions `filter`, `map` and `reduce` are doing.

3 Distributed least squares

We consider the the least squares problem

$$\min_{w\in\mathbb{R}^{d\times 2}} \frac{1}{2}\|Xw_{:,\text{ethylene}} - y_{:,\text{ethylene}}\|^2 + \frac{1}{2}\|Xw_{:,\text{methane}} - y_{:,\text{methane}}\|^2. \tag{1}$$

Question 3.1
Show that the solution to the least squares problem is the solution to a linear system of equations of a moderate size.

Question 3.2
Solve the least squares problem

$$\min_{w\in\mathbb{R}^{d\times 2}} \frac{1}{2}\|Xw_{:,\text{ethylene}} - y_{:,\text{ethylene}}\|^2 + \frac{1}{2}\|Xw_{:,\text{methane}} - y_{:,\text{methane}}\|^2.$$

where X is the reduced-size training data, stored as an array in the variable `trainData_NotDistributed`.

Question 3.3
Solve with parallel computations the least squares problem

$$\min_{w\in\mathbb{R}^{d\times 2}} \frac{1}{2}\|Xw_{:,\text{ethylene}} - y_{:,\text{ethylene}}\|^2 + \frac{1}{2}\|Xw_{:,\text{methane}} - y_{:,\text{methane}}\|^2.$$

You should compute every sum over the observations using Spark's `reduce` function.

Run the algorithm on the reduced-size data and on the full training data.

4 Quantile regression with linear kernels: serial algorithm

For $\tau \in (0,1)$, let us consider the pinball loss defined as $L_\tau(v) = \max\{-(1-\tau)v, \tau v\}$ and the function f_2 such that $f_2(v) = L_\tau(y - v)$.

Question 4.1
Calculate $\text{prox}_{\gamma f_2}(v)$ for $\gamma > 0$.

In our dataset, we have d features and n observations. Given $\tau \in (0,1)$, we would like to estimate the τ conditional quantile of the gaz concentrations $y_{:,\text{ethylene}}$ and $y_{:,\text{methane}}$ knowing X. In that purpose, we solve the following optimization problem

$$\min_{w\in\mathbb{R}^{d\times 2}, w_0\in\mathbb{R}^2} \frac{\alpha}{2}\sum_{g=1}^{2}\sum_{j=1}^{d} w_{j,g}^2 + \sum_{g=1}^{2}\sum_{i=1}^{n} L_\tau\big(y_{i,g} - \sum_{j=1}^{d} x_{i,j}w_{j,g} - w_{0,g}\big) \tag{2}$$

where $\alpha > 0$ is a regularization constants. We shall take $\tau = 0.9$ and $\alpha = 1$.

We split the quantity $L_\tau\big(y_{i,g} - x_i^\top w_{:,g} - w_{0,g})\big)$ which is neither smooth nor differentiable and we obtain the problem

$$\min_{w\in\mathbb{R}^{d\times 2}, w_0\in\mathbb{R}^2, z\in\mathbb{R}^{n\times 2}} \frac{\alpha}{2}\|w\|_F^2 + \sum_{g=1}^{2}\sum_{i=1}^{n}\Big(L_\tau\big(y_{i,g} - z_{i,g}\big) + \iota_{\{0\}}(x_i^\top w + w_0 - z_{i,:})\Big) \tag{3}$$

Note that the new variables $z_{i,:}$ are naturally distributed across the computing agents and so the processor dealing with observation i should also deal with the variable $z_{i,:}$.

We first consider a serial algorithm dealing with centralized data. We will focus on the small training data stored as the numpy array `trainData_NotDistributed`.

If you failed to have Spark running, you may complete this section with the following data:

```
data = np.loadtxt(filename, skiprows=1)
train_data_NotDistributed = data[:data.shape[0] // 2:100]
```

Question 4.2
Write Problem (3) as

$$\min_{w,w_0,z} f_1(w,b) + f_2(z) + \iota_{\{0\}}(\mathcal{M}(w,w_0) - z)$$

where you specify the functions f_1 and f_2 and the matrix $\mathcal{M}$.

Question 4.3
Write the ADMM for solving this quantile regression problem.

Question 4.4
Specify the algorithms necessary to solve each step of the ADMM.

Question 4.5
Implement the optimization algorithm. Run it with initial point $(z,\mu) = 0$ and parameter $\gamma = 1$. Set the iteration limit to 100 and monitor the norm of $\mathcal{M}(w,b) - z$ and the value $f_1(w,b) + f_2(z)$.

5 Parallel algorithm

Question 5.1
Adapt the algorithms written in the serial centralized case for distributed data.

You may wish to define a distributed data structure containing the distributed optimization variables as

```
train_data_with_z_and_mu = train_data.map(lambda row: {'y' : row[1:3],
                                                       'x' : row[3:],
                                                       'z' : np.zeros(2),
                                                       'mu' : np.zeros(2)})
train_data_with_z_and_mu.cache()
```

Question 5.2
Run the serial and the parallel algorithm on the small training set. Compare the solutions you obtain and the training times with 100 iterations of the ADMM.

Question 5.3
Run the parallel algorithm on the full training set for 20 iterations.

Question 5.4
Compare the test errors you obtain with the small and the large training sets.