Computer lab :
Let's reverse-engineer the data center
SD-TSIA 211

Olivier Fercoq
January 2024

1 Submission and grading information

You can do the computer lab alone or in pairs. Please write a report and post it on `e-campus`. You can do it as a jupyter notebook or a pdf file.

Then, each of you will have to evaluate a couple of other students' reports and give comments.

Only the fact that you produce a report and evaluate your peers count in the final grade, so do not worry if you do not finish everything.
- 1 point for being present on the day of the lab
- 1 point for submitting a report
- 1 point for commenting 2 reports

2 Database and statistical model

Welcome to Paname Télécom ! You have been recently hired by this young company that would like to become a challenger in the competitive field of data centers. In order to gain quick knowledge on how to run a data center efficiently, our team of hackers has used a security breach to steal some data from a well known data center. On `e-campus`, you will be able to download the files `data_center_data_matrix.npy` and `data_center_helper.py` that contain the data and basic operations that can be done on them.

The data is composed of measurements with a roughly two-hour sampling rate together with 4 key performance indicators (KPIs). However, we were not able to get the formula that gives the indicators as a function of the data. Your mission is thus to reverse-engineer the performance indicator. We conjecture that they can be written as a ratio of affine transforms of the raw data, subject to some noise.

This gives the following model for the KPI number i at time t :

$$y_i(t) = \frac{w_{i,1}^\top x(t) + w_{i,0} + \epsilon_i(t)}{w_{i,2}^\top x(t) + 1}$$

where $x(t) \in \mathbb{R}^d$ is the list of all the measurements at time t, $\epsilon_i(t)$ is an i.i.d. noise and $(w_{i,0}, w_{i,1}, w_{i,2}) \in \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d$ are the parameters of the model.

We shall concentrate on the KPI number 3 to start with and simplify $(w_{i,0}, w_{i,1}, w_{i,2})$ into (w_0, w_1, w_2) since i will be always 3.

As is usual in machine learning, we split the data into two parts : $(x(t), y(t))_{t\in\{0,\dots,N_{train}-1\}}$ for training and $(x(t), y(t))_{t\in\{N_{train},\dots,N_{train}+N_{test}-1\}}$ for testing.

3 Least squares

In order to fit this model to the data, we first standardize the data. Indeed, the measurements may have various units (like kWh, degree Celsius, V, etc) and it makes more sense to separate the statistical aspects from these dimensionality considerations. Hence, we consider the matrix $\tilde{x}$, which is such that each of its columns has mean and standard deviation respectively 0 and 1 over the training set. (Note that we do not use the test set to do this standardization.)

We then solve the following least squares problem

$$\min_w \frac{1}{2}\|Aw - b\|^2$$

where $(Aw)_t = \tilde{x}(t)^\top w_1 + w_0 - y(t) \times \tilde{x}(t)^\top w_2$ for all t and $b_t = y(t)$.

Question 3.1

Show that if $Aw = b$, then $y(t) = \dfrac{w_1^\top \tilde{x}(t) + w_0}{w_2^\top \tilde{x}(t) + 1}$.

Question 3.2

Solve this least squares problem using the function `numpy.linalg.lstsq` .

Question 3.3

Evaluate the quality of the solution found on the test set, by computing the mean-squared error

$$\frac{1}{N_{\text{test}}}\|A_{\text{test}}w - b_{\text{test}}\|_2^2.$$

Question 3.4

In order to improve the generalization power of the model, we consider a ℓ_2 regularization :

$$\min_w \frac{1}{2}\|Aw - b\|^2 + \frac{\lambda}{2}\ \|w\|^2$$

where $\lambda = 10^4$.

Calculate the gradient of $f_1 : w \mapsto \frac{1}{2}\|Aw - b\|^2 + \frac{\lambda}{2}\|w\|^2$. Is the function convex ?

Question 3.5

Implement gradient descent to minimize f_1. What step size are you choosing ? How many iterations are needed to get w_k such that $\|\nabla f(w_k)\| \leq 1$?

Question 3.6
Compare the test mean squared error with the solution obtained without regularization.

Question 3.7
(Bonus question) Implement the accelerated gradient method. Compare the convergence speed with the previous algorithm.

4 Regularization for a sparse model

You may have seen that at the optimum, the parameter w has many coordinates with small but nonzero values. In order to force most of them to be exactly 0 and thus, to concentrate on the really informative features, we can solve a Lasso problem, that is a least squares problem with ℓ_1 regularization :

$$\min_w \frac{1}{2}\|Aw - b\|^2 + \lambda \ \|w\|_1$$

Question 4.1
Write the objective function as $F_2 = f_2 + g_2$ where f_2 is differentiable and the proximal operator of g_2 is easy to compute. Recall the formula for prox_{g_2}. Calculate the gradient of f_2.

Question 4.2
Code the proximal gradient method. Here, we will take $\lambda = 10^3$. What stopping criterion do you suggest ?

Question 4.3
Compute the test error. Plot the values of the solution obtained : what do you observe ? Compare with the one obtained with ℓ_2 regularization.

Question 4.4
We may try to accelerate the algorithm using line search. Compare the speed of the algorithm with fixed step size and with line search.

5 Choice of the regularization parameter

You may not have time to code this part of the computer lab but it may be worth understanding what can be done in order to choose the regularization parameter.

A natural question when considering a regularized machine learning problem is : what is the best value for the regularization parameter ρ ? Its goal is to force the model to choose less complex solutions in order to generalize better.

Hence, to evaluate the generalization performance, we are going to split our data into a training set X_{train}, y_{train} and a validation set X_{valid}, y_{valid}. Then, we solve the regression problem using the training set but test its performance on the validation set. Note that the

loss function for the validation set is not necessarily the mean square error (but we will keep it as the mean square error here).

Gathering everything the problem we are trying to solve is the following bilevel optimization problem

$$\min_{\rho \geq 0} \frac{1}{n_{\text{valid}}} \sum_{j=1}^{n_{\text{valid}}} f(w^{(\rho)}, x_j, y_j)$$
$$\hat{w}^{(\rho)} \in \arg\min \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} f(w, x_i, y_i) + \rho R(w)$$

where $f(w, x_i, y_i) = \frac{1}{2}\|Aw - b\|^2$ and $R(w)$ is either $R(w) = \frac{1}{2}\|w\|_2^2$ or $R(w) = \|w\|_1$.

Since this is a complex nonconvex optimization problem, we are going to evaluate the accuracy on a grid of values for ρ, that is $\rho \in \{\rho_0 a^k : k \in \{0, 1, \ldots, K\}\}$ for given $\rho_0 > 0$, $0 < a < 1$ and K. Then, we select the parameter $\hat{w}^{(\rho)}$ that has the smallest MSE loss on the validation set.

6 Comparison

Question 6.1
Compare the solutions obtained by the two types of regularization with different regularization parameters. What is the best solution ?