Optimization for Machine Learning - Final Project

Avigail Zerrad

M2 MASH, 2020-2021

Université Paris-Dauphine – MIDO –, Place du Maréchal de Lattre de Tassigny, 75016 PARIS

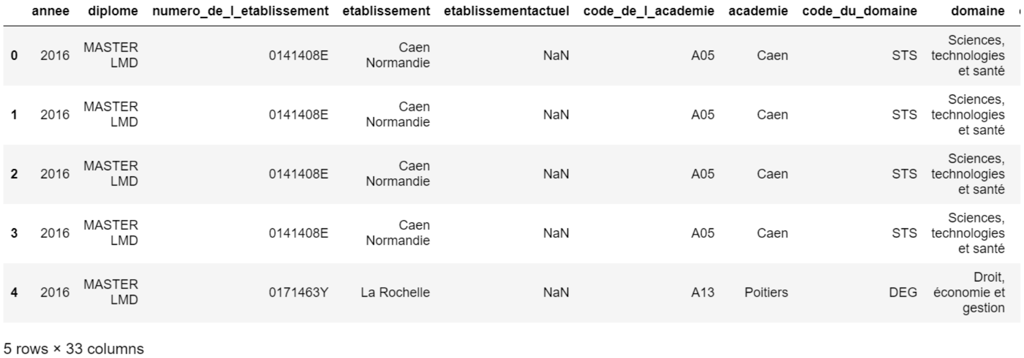
Contents

Introduction

This report presents the results obtained for the Final Project of the course Optimization for Machine Learning. I mostly used the code from the 'Lab session on SGD' by Clément Royer.

1 Presentation of the dataset

The dataset I decided to use is from the website **data.gouv.fr**. You can find this open data here. This information is based on data collected as part of the national data collection operation on the professional integration of Master's graduates.



	annee	diplome	numero_de_l_etablissement	etablissement	etablissementactuel	code_de_l_academie	academie	code_du_domaine	domaine
0	2016	MASTER LMD	0141408E	Caen Normandie	NaN	A05	Caen	STS	Sciences, technologies et santé
1	2016	MASTER LMD	0141408E	Caen Normandie	NaN	A05	Caen	STS	Sciences, technologies et santé
2	2016	MASTER LMD	0141408E	Caen Normandie	NaN	A05	Caen	STS	Sciences, technologies et santé
3	2016	MASTER LMD	0141408E	Caen Normandie	NaN	A05	Caen	STS	Sciences, technologies et santé
4	2016	MASTER LMD	0171463Y	La Rochelle	NaN	A13	Poitiers	DEG	Droit, économie et gestion

5 rows × 33 columns

Figure 1: Dataset

The original dataset has 33 columns. However, some variables are redundant ('etablissement'/'code de l'etablissement'; 'code de la discipline'/'discipline'...) or not informative. That's why, I decided to keep only some features. These variables make sense for the task I chose (predict the salary).

Moreover, the dataset contains 13 797 rows, but the documentation associated to the data specifies that some information are not available ('nd') or non significant ('ns'). I decided to remove the rows with those missing values.

1.1 Data vizualization

Heatmap I used an heatmap to see correlations between numerical features.

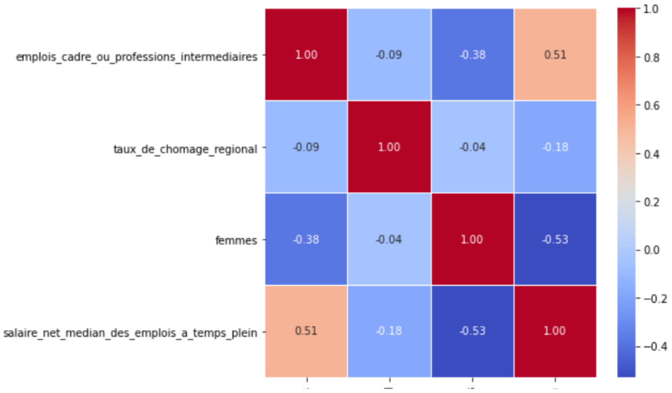


Figure 2: Heatmap for numerical features

We notice that :

- 'emplois cadre ou professions intermediaires' and 'salaire net median des emplois a temps plein' are positively correlated. It makes sense, because the salary increases with the status.
- 'femmes' and 'salaire net median des emplois a temps plein' are negatively correlated, meaning that more we have women, less the salary is important. A good representation of this sad reality...

Histograms On the histograms of the numerical variables (Figure 3), we can observe that :

- 'emplois cadre ou professions intermediaires' : There is a peak on the right, meaning that most students are employed in a managerial/intermediate job at the end of their Master's degree.
- 'salaire net median des emplois a temps plein': There are two peaks (1800 and 2000), meaning that a lot of fields will lead to a median salary around 1800, and a lot of fields will lead to a median salary around 2000.

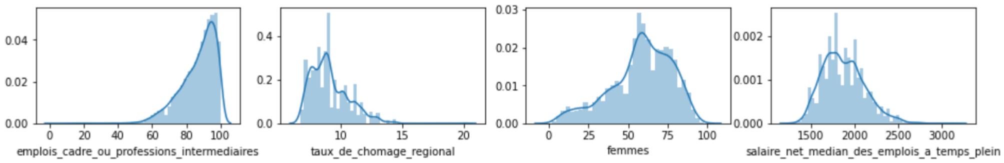


Figure 3: Histograms for numerical features

1.2 Data preprocessing

One of the most issues I had with the dataset I chose is the presence of categorical variables. For example, the feature 'discipline' contains 20 modalities, the feature 'etablissement' contains 90 modalities... As we can see, these features are not ordinal, so we should encode them using a one-hot-encoder. However, it leads to a dataset with a lot of columns, and the running of the methods is then very slow. To avoid this problem, I encoded the categorical features using LabelEncoder from sklearn, even though it does not make a great sense for the data.

2 Presentation of the problem

Now that the data is ready, we can focus on the problem. The goal is to make a linear regression in order to predict the median salary.

We will use different methods from the course in order to minimize the following objective function :

$$\min_{w \in R^d} f(w) := \frac{1}{2n} \|Xw - y\|^2$$

Here, X is the design matrix (which contains the features) and y is the vector of median salaries.

3 Batch gradient descent

We run Gradient Descent with a constant stepsize $\alpha_k = \frac{1}{L}$, where L is a Lipschitz constant for ∇f, and with a decreasing stepsize $\alpha_k = \frac{\alpha_0}{\sqrt{k+1}}$, where α_0 is an input parameter.

We obtain the following figure :

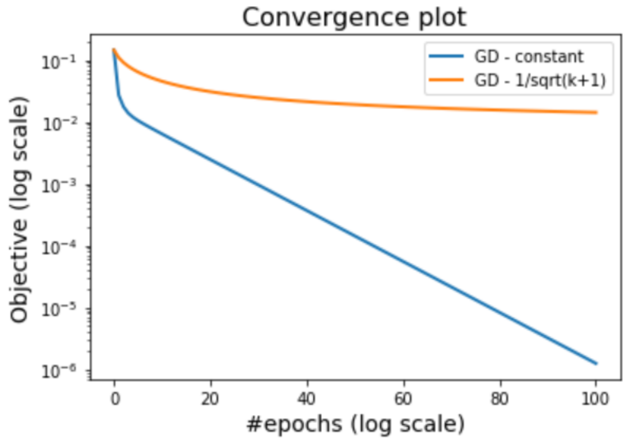


Figure 4: GD - Constant vs decreasing stepsize

Link with the course We saw at many times that, for a smooth and strongly convex function, convergence theorem shows that linear rate is obtained when we run gradient descent with constant stepsize. We can see this linear rate thanks to the logscale (see Figure 4). In terms of speed of convergence, it means that the convergence to the solution is exponentially fast.

4 Accelerated gradient descent (Nesterov)

I implemented Nesterov's acceleration using the definition from the course of Irène Waldspurger, that is :

$$\begin{gathered} x_{t+1} = y_t - \frac{1}{L}\nabla f\left(y_t\right) \\ y_{t+1} = x_{t+1} + \gamma_t\left(x_{t+1} - x_t\right) \end{gathered}$$

where $(\gamma_t)_t$ is defined as following :

$$\begin{gathered} \lambda_{-1} = 0 \\ \forall t \in N, \quad \lambda_t = \frac{1+\sqrt{1+4\lambda_{t-1}^2}}{2} \\ \forall t, \quad \gamma_t = \frac{\lambda_t - 1}{\lambda_{t+1}} \end{gathered}$$

We then obtain :

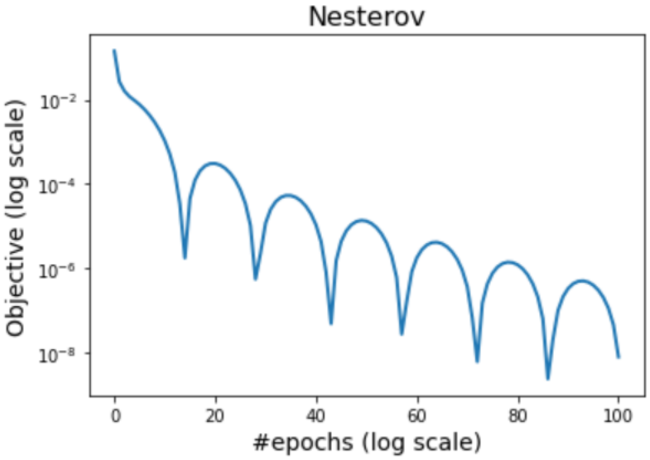


Figure 5: Nesterov acceleration

5 Stochastic gradient descent

The iteration of Stochastic Gradient Descent is given by:

$$w_{k+1} = w_k - \alpha_k \nabla f_{i_k}(w_k),$$

where i_k is drawn at random in $\{1, \dots, n\}$.

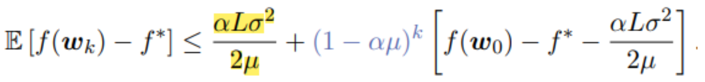
5.1 Learning rate

Running SGD with the constant stepsize $\alpha_k = \frac{1}{L}$ shows that the algorithm does not converge.

Link with the course In the course on SGD of Clément Royer, we saw that for SGD with constant stepsize, 'residual terms appear in the convergence rate, that are related to the variance of the gradient estimator'.

Then, SGD with constant stepsize diverges because the noise prevents from guaranteeing the difference with the optimal value (in expectation) remains finite. We can see it

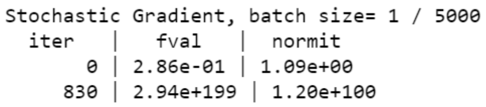
- in theory (the term in yellow corresponds to the noise) :



$$\mathbb{E}\left[f(\boldsymbol{w}_k) - f^*\right] \leq \frac{\alpha L \sigma^2}{2\mu} + (1 - \alpha\mu)^k \left[f(\boldsymbol{w}_0) - f^* - \frac{\alpha L \sigma^2}{2\mu}\right].$$

Figure 6: Theorem 2.3.1 - SG with constant stepsize

- in practice :



```
Stochastic Gradient, batch size= 1 / 5000
  iter    |    fval     |   normit
       0  | 2.86e-01  | 1.09e+00
     830  | 2.94e+199  | 1.20e+100
```

Figure 7: Divergence of SGD with constant stepsize

We then use the decreasing stepsize $\alpha_k = \frac{\alpha_0}{\sqrt{k+1}}$, and obtain the Figure 8 :

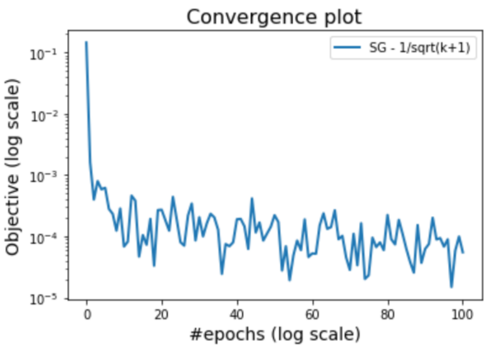


Figure 8: SGD - Decreasing stepsize

Link with the course In the course on SGD of Clément Royer, Theorem 2.3.2 leads to the conclusion that 'choosing a decreasing stepsize results in a sublinear convergence rate'. We can indeed observe it on Figure 8.

5.2 (Mini-)Batch size

Formally, the update of a batch stochastic gradient method is given by

$$w_{k+1} = w_k - \frac{\alpha_k}{|S_k|} \sum_{i \in S_k} \nabla f_i(w_k)$$

where S_k is a set of indices drawn uniformly in $\{1, \ldots, n\}$.

The following figure presents the results obtained after running gradient descent, stochastic gradient descent and mini-batch stochastic gradient with different batch sizes.

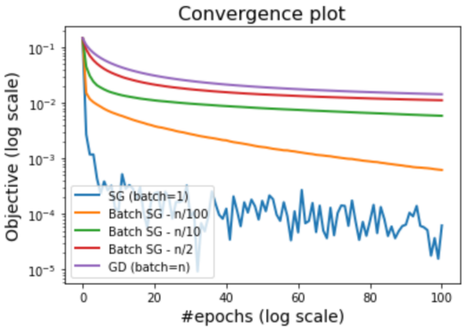


Figure 9: Mini-batch SG

We observe that mini-batch SG is faster than gradient descent and performs some variance reduction compared to SGD.

5.3 Diagonal scaling

Stochastic gradient is not scale invariant. A way to fix it is to use diagonal scaling. Two approaches, RMSProp and Adagrad, are illustrated here.

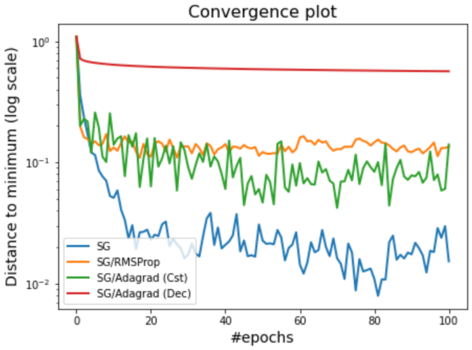


Figure 10: SGD - Diagonal scaling

These results don't really improve what we already had.

5.4 Iterate averaging

We saw that mini-batch SG is a good way to reduce variance. Another way is to use Iterate averaging. The figure 11 shows the obtained results :

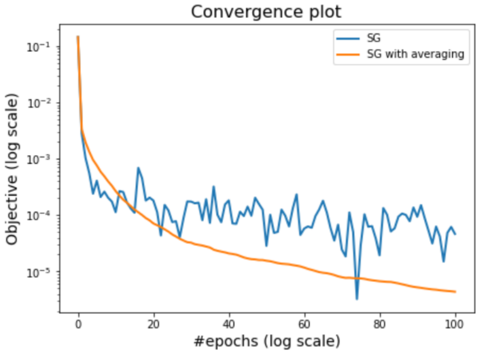


Figure 11: Iterate averaging

We observe that the use of this method is a good way to limit the oscillating behavior of stochastic gradient.

5.5 SAGA

Iterate averaging showed a sub-linear rate. We see in this section that the gradient aggregation method named SAGA, leads to a linear convergence rate. It is shown by the Theorem 3.1.2 of the course, and we observe it on the Figure 12.

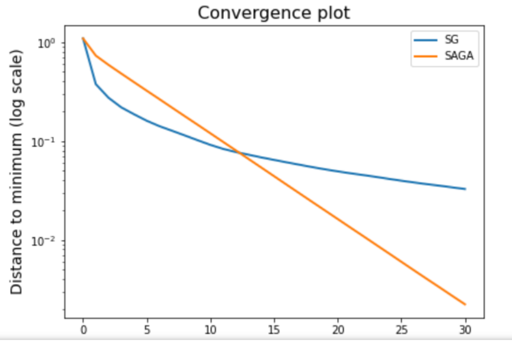


Figure 12: SAGA

6 Regularization

6.1 Ridge

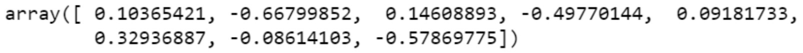
The goal is to solve

$$\min_{w \in R^d} f(w) := \frac{1}{2n}\|Xw - y\|^2 + \lambda \|w\|_2^2.$$

The Proposition 8 from the course of Gabriel Peyré shows that we have :

$$w_\lambda = \left(X^\top X + \lambda \mathrm{Id}\right)^{-1} X^\top y$$

For $\lambda = 0$, we obtain :

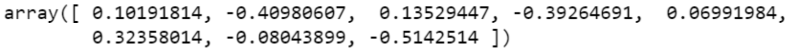


```
array([ 0.10365421, -0.66799852,  0.14608893, -0.49770144,  0.09181733,
        0.32936887, -0.08614103, -0.57869775])
```

Figure 13: $\lambda = 0$

We observe that the variables x5 ('situation', i.e beeing at 18 or 30 months after the end of the studies at the moment of the survey) and x7 ('taux de chomage regional') seem to be the less informative variables for our task.

For $\lambda = 100$, we obtain :



```
array([ 0.10191814, -0.40980607,  0.13529447, -0.39264691,  0.06991984,
        0.32358014, -0.08043899, -0.5142514 ])
```

Figure 14: $\lambda = 100$

We observe that Ridge regularization has not a great impact on the variables, except maybe for the features x2 ('diplome', i.e Master LMD or ENS) and x4 ('discipline'). We can see it graphically (Figure 15).

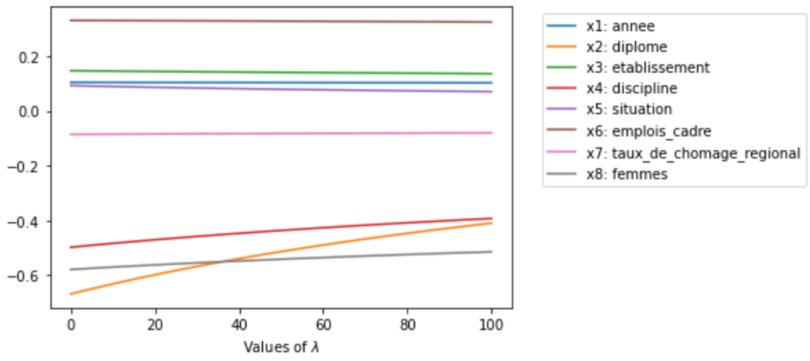


Figure 15: Ridge

A good way to evaluate the obtained estimator is to compute the Mean Squared Error.

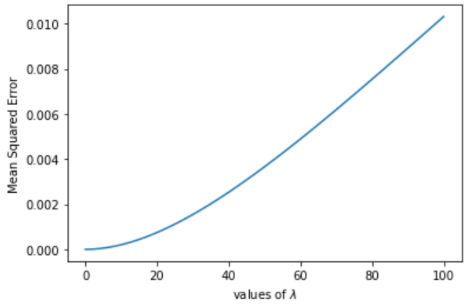


Figure 16: MSE - Ridge

We can then conclude that the best λ is $\lambda = 0$, which corresponds to the initial problem without regularization.

6.2 Lasso using iterative soft thresholding

In order to perform feature selection, we have to replace the $l2$ regularization penalty by a sparsity inducing regularizer. The most well known is the $l1$ norm. Then, we want to solve the Lasso problem:

$$\min_{w \in R^d} f(w) := \frac{1}{2n} \|Xw - y\|^2 + \lambda \|w\|_1.$$

.

We use the algorithm of Iterative Soft Thresholding, defined by :

$$w_{k+1} = \mathcal{S}_{\lambda\tau} \left(w_k - \tau X^\top (Xw_k - y) \right)$$

where :

- $\mathcal{S}_\lambda(r) = \max(|r| - \lambda, 0) sign(r)$
- τ is such that $0 < \tau < 2/\|X\|^2$ to ensure convergence.

We obtain the following figure by taking $\tau = 1/\|X\|^2$:

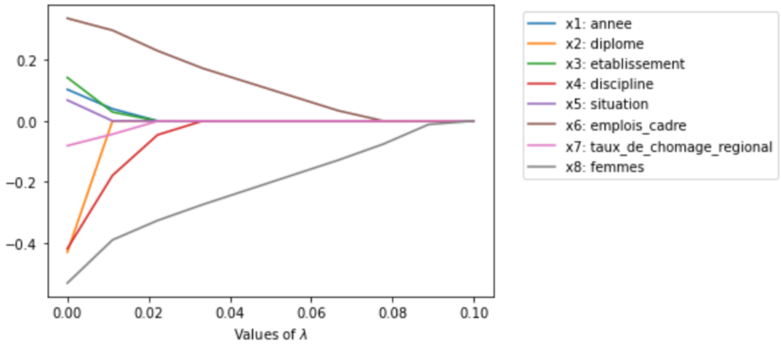


Figure 17: Lasso

We observe that the most significant variables for our task are the variables x8 ('femmes'), x6 ('emplois cadre') and x4 ('discipline'). Indeed, they are the last features to be thresholded to zero. These results are consistent and make sense in our context.

Conclusion

This project was a good way to experiment the theorical results we saw in course. I particularly appreciated the correspondence between the convergence rates demonstrated during the lectures and the graphs that showed it in practice. About the regularization, Ridge didn't seem to be adapted to the data, but Lasso showed good results for feature selection.