Comparison between some regularization techniques in Least Square Regression and Logistic Regression

Seyni DIOP*, Alhousseynou BALL*, and Ndeye LO*

*Students at M2 Datascience - Ecole Polytechnique X

4th March 2020

Contents

1 Introduction

In supervised learning models, learning is done through a training set. However, our goal is not that our model works well on our training set but rather that it can be generalized to other unobserved data.

However, it happens very often that after a certain amount of learning, the model starts to lose its generalization power: we talk about overfitting. It arrives when the model is beginning to learn signal as well as noise in the training data and wouldn't perform well on new data on which model wasn't trained on.

"Regularization is the process of adding information in order to solve an ill-posed problem or to prevent overfitting."[1]

In this report we will study and compare regularization techniques such as Ridge regularization, Averaging Stochastic Gradient Descent(one pass), Multiple Passes Stochastic Gradient Descent and early stopping in the particular case of linear regression and logistic regression.

2 Overview about some regularization techniques

In this section we will try to detail some regularization techniques such as the ridge, give the mathematical formulas and also explain a little bit how stochastic methods can be seen as regularization methods, etc.

Problem

The problem here consists in minimizing a certain loss function (half of the mean square error (MSE) for instance in linear regression)

$$\min \left\{ \mathcal{L}(\beta) := \frac{1}{n} \sum_{i=1}^{n} \ell(x_i^\top \beta, y_i) \right\} \tag{1}$$

where we consider a learning setting given by a probabilistic space $(X \times Y, \rho(X, Y)), Y \in \mathbb{R}$ and $S = \{x_i, y_i\}_{i=1}^n$ denote a training set of n pairs i.i.d. with respect to ρ.

Suppose that $x_i \in \mathbb{R}^d$

Linear Regression

In linear regression, the loss function is :

$$\mathcal{L}(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - x_i^\top \beta)^2 \tag{2}$$

Logistic Regression

[1] https://en.wikipedia.org/wiki/Regularization_(mathematics)

The logistic regression algorithm used for multi-classification task is called multinomial logistic regression.

In multinomial logistic regression, the label y can take K different values ($K \geq 2$). Thus, in our training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, we now have that $y_i \in \{1, 2, \ldots, K\}$.

Given a test input x, we want our hypothesis to estimate the probability that

$\mathbb{P}(y = k|x_i; \beta)$ for each value of $k = 1, \ldots, K$:

$$\mathbb{P}(y_i = k|x_i; \beta) = \frac{\exp(x_i \beta_k)}{\sum_{j=1}^{K} \exp(x_i \beta_j)}$$

In multinomial logistic regression, we have cross entropy loss defined by:

$$\mathcal{L}(\beta) = -\sum_{i=1}^{m} \sum_{k=0}^{1} \mathbb{1}\{y_i = k\} \log \mathbb{P}(y_i = k|x_i; \beta) \tag{3}$$

2.1 Ridge regularization

In Ridge Regression, the loss function is augmented in such a way that we not only minimize the simple loss but also penalize the size of parameter estimates (β in notation), in order to shrink them towards zero. And the ridge regression can be formalize by:

$$\min \left\{ \mathcal{L}(\beta, \lambda) := \frac{1}{n} \sum_{i=1}^{n} \ell(x_i^\top \beta, y_i) + \frac{\lambda}{2} ||\beta||^2 \right\} \tag{4}$$

with λ in $\mathbb{R}_+$.

NB: λ is called regularization penalty. We can note notice if λ goes to 0 we obtain the simple problem1.

We can determine analytically the solution of problem 4 in a case of linear regression , i.e. $\ell(x_i^\top \beta, y_i) = \frac{1}{2}(y_i - x_i^\top \beta)^2$ and we obtain

$$\hat{\beta}^{ridge} = (\mathbf{X}^\top \mathbf{X} + n\lambda I)^{-1} \mathbf{X}^\top \mathbf{Y} \tag{5}$$

where $\mathbf{X} = [x_1, x_2, ..., x_n] \in \mathbb{R}^{n \times d}$ and $\mathbf{Y} = (y_1, y_2, ..., y_n) \in \mathbb{R}^n$. And under the notation, the β^{ridge} can be calculated with $\mathcal{O}(nd^2)$ [2].

We can see that the solution to the minimization problem depends on λ. Legitimate questions would then be: *"When do we have to regulate by ridge? Does ridge regulation bring something for any situation? If so, what should the λ scale be? Doesn't it depend on the size of the training set?"*

[2] see appendix A for details

2.2 Stochastic Gradient Descent Methods

What's Gradient Descent ?

We also know that a function reaches its minimum when its derivative is equal to 0, using the following fixed point type algorithm (called gradient descent algorithm)

$$\beta^{t+1} = \beta^t - \gamma_{t+1} \nabla \mathcal{L}\left(\beta^t\right) \tag{6}$$

where γ_t is the learning rate at iteration t.

> *Gradient descent is an iterative algorithm, that starts from a random point on a function and travels down its slope in steps until it reaches the lowest point of that function.*[3]

This technique and its variants are used in many areas of supervised learning. GD ensures the reduction of the learning error on the training set but not on the test set(with a good learning rate). And as explained earlier in the introduction; during the learning phase and therefore after a certain number of iterations the β solution starts to lose its generalization power, this is over-fitting.

In addition to overfit the training data after a certain number of iterations, gradient descent is costly in the sense that one iteration costs $\mathcal{O}(nd)$.

Stochastic methods have been developed at least by Robbins and Monro (Robbins and Monro (1951)). The regularizing properties of SGD remains to be established, and it was conjectured and empirically examined in Jastrzębski et al. (2017) and Keskar et al. (2017). It is shown that under certain assumptions (*smoothness, convexity of the loss function, etc.*), stochastic methods could be seen as a Tikhonov regularization method (ridge) (Lin et al. (2016), Jin and Lu (2018)). And the regularizer in this case, the SGD can be seen as a Tikhonov regularization by controlling the number of passes on the data and/or the learning step-size (Lin et al. (2016)).

But also for large problems, it is still preferable to have optimal parameters or more or less in a region close to optimality (for generalization) by just doing a pass on the data. The asymptotic performance of the SGD by averaging the parameters has been established by Boris T. Polyak and Anatoli. B. Juditsky. (1992).

In our case we will update the parameters β with just a 1-sample set taken randomly from $S = \{x_i, y_i\}_{i=1,..,n}$.

We're going to present two variants of SGD for regularization:

- Multi-passes SGD
- One pass SGD + averaging

[3] https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31

Averaging Stochastic Gradient Descent

For *one pass* we take $T = n = sample\ size$

1. initialization $\beta^0 = 0$, choose $\alpha_t > 0$
2. for $t = 0, 1, 2, \ldots, T-1$
   (a) sample $j \in \{1, \ldots, n\}$
   (b) $\beta^{t+1} = \beta^t - \gamma_{t+1} \nabla \mathcal{L}_j \left(\beta^t\right)$
   (c) if $t > s_0$
   $\quad \bar{\beta} = \frac{1}{t+1-s_0} \sum_{i=s_0}^{t} \beta^i$
   else : $\bar{\beta} = \beta^{t+1}$
3. output $\bar{\beta}$

Multiple passes SGD

Define by P the number of passes to do

1. initialization $\beta^0 = 0$, choose $\gamma_t > 0$
2. for $t = 0, 1, 2, \ldots, nP-1$
   (a) sample $j \in \{1, \ldots, n\}$
   (b) $\beta^{t+1} = \beta^t - \gamma_{t+1} \nabla \mathcal{L}_j \left(\beta^t\right)$
3. output β^{nP}

2.3 Early stopping

It is ideally hoped that our model can be generalized; what our model formed from the gradient descent loses over a number of iterations. A natural alternative to regularization is to stop learning at certain moments to avoid overfitting : it's early stopping. This main advantage is generally his lower computational complexity compared to penalized forms like ridge (Raskutti et al. (2014)).

OK early stopping ! but when ?

Suppose that the data $S = \{x_i\}_{i=1}^n$ are subdivided into two sub-samples Str and Ste. The data indexed by the Str set are used to estimate the model parameters using the gradient descent update. At each iteration $t = 0, 1, 2, ...$ the data indexed by Ste are used to estimate the prediction error via $\mathcal{L}_{Ste,t}$. The Hold Out stopping method defines the following stopping rule

$$\hat{T}_{HO} = \arg\min \left\{ t \in \mathbb{N} \quad | \quad \mathcal{L}_{Ste,t+1} > \mathcal{L}_{Ste,t} \right\} - 1 \tag{7}$$

The stopping criterion presented above is the simplest, however other more sophisticated empirical stopping rules can be considered (Prechelt (2014)) but also theoretical boundaries (Raskutti et al. (2014)).

3 Data and Results

3.1 Data Presentation

The data used comes from UCI Machine Learning

- *for mean squared regression*: a dataset with variables concerning the construction of buildings will be used. It will be question to predict the variables the cost of building construction. Our predictors (variables used in the regression) of the variables related to the geographical location of the building, its size, the duration of construction and many other variables.[4]
- *for logistic regression*: the goal is to map different forest types using spectral data in Japan. The different values taken by the target variable are : s('Sugi' forest), 'h' ('Hinoki' forest), 'd' ('Mixed deciduous' forest), 'o' ('Other' non-forest land) .[5]

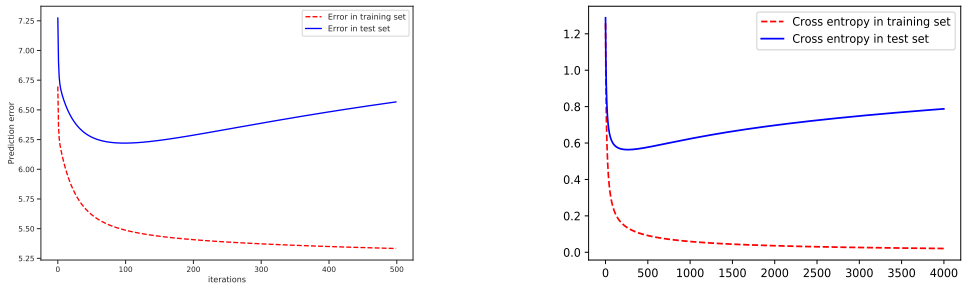
Table 1: Data description

	Instances	**Atributes**	**Missing Values**
Residential Dataset	*372 (25% for test)*	*105*	-
Forest Type Mapping Data Set	*train : 198* *test : 325*	*27*	-

The data were standardized before being used. A test set of 25% in original data was randomly drawn for the residential dataset. And for the forest type mapping we just used the test base provided.

3.2 Results

We will use as an error function to minimize the MSE for the case of least squares regression and the cross entropy for the case of logistic regression.



(a) Prediction error: Residential dataset (b) Cross entropy : Forest mapping dataset

Figure 1: Overfitting highlight - Gradient descent

[4] https://archive.ics.uci.edu/ml/datasets/Residential+Building+Data+Set
[5] https://archive.ics.uci.edu/ml/datasets/Forest+type+mapping

In the figure on the left we can see an increase in the prediction error(mse here) from a certain number of iteration(figure 1a). And by doing with the optimal parameters for the linear regression problem (problem 1), we obtain a prediction error of 7970.638 in the test set. Similarly, the cross entropy tends to increase during the training (figure 1b) and logistic regression done with sklearn give 3.1686.

3.2.1 About ridge

In this part we will study the case of ridge regularization on our datasets.

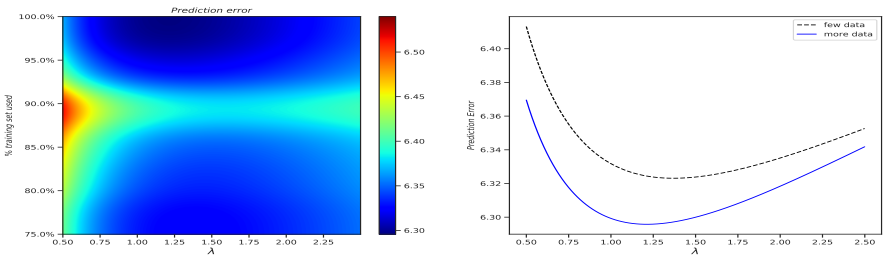


Figure 2: Impact of regularizer(least squares linear regression- Residential dataset)

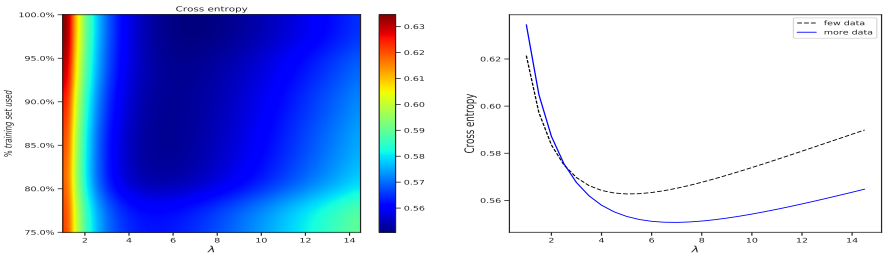
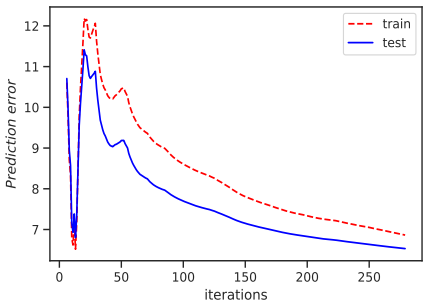


Figure 3: Impact of regularizer(logistic regression - Forest type dataset)

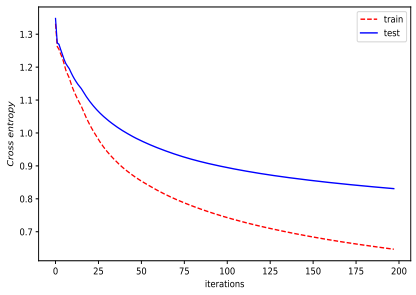
Here we note a clear improvement of our regularized model and we can also see that the choice of the hyperparameter λ is crucial. Moreover it is to note that the more data we have, the weaker the regularization must be (λ smaller) and the better the generalizations are.

3.2.2 About Stochastic Gradient Descent Methods

1. Averaging Stochastic Gradient Descent(ASGD)(*one pass*)



(a) Prediction error: Residential dataset

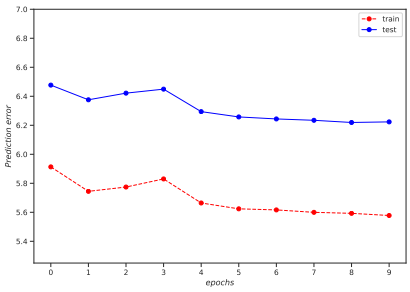


(b) Cross entropy during training: Forest mapping dataset

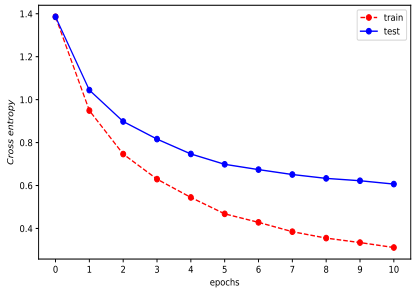
Figure 4: *One Pass Averaging*

We chose learning $\gamma_t = \gamma_0(1 + a\gamma_0 t)^{-c}$ with $\gamma_0 = 1$, $a = \lambda_0(\text{max of eigenvalues of } \mathbf{X})$ and $c = 2/3$ for linear regression and $c = 3/4$ for logistic regression as suggested by Xu (2011) allowing the averaging stochastic gradient descending to reach an asymptotic region.$(t > \mathcal{O}(\lambda_0\gamma_0)^{-1}))$

2. Multiple passes Stochastic Gradient Descent



(a) Prediction error: Residential dataset



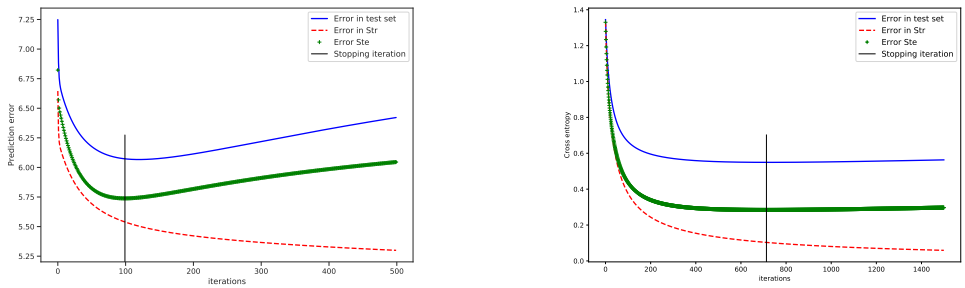
(b) Cross entropy during training: Forest mapping dataset

Figure 5: *Multiple passes SGD*

In this point we perform a simple SGD (1-sample) but over many passes. The learning rate used is $\gamma_t = \gamma_0 t^{-\alpha/(\alpha+1)}$ with $\gamma_0 = 1/L$ and $\alpha = 1$ (Lin et al. (2016), Tong Zhang (2004)) A constant learning rate could be considered as having sometimes good properties when you want to make a small number of passes on the data. However, in this case, the learning rate must be properly chosen (Lin et al. (2016)). The learning rate must be not too small and also not too big to avoid ocscillations and divergence[6].

[6] see Appendix B for comparaison

3.2.3 About early stopping



(a) Prediction error: Residential dataset

(b) Cross entropy during training: Forest mapping dataset

Figure 6: *Early stopping*

Here we split our training set in two sets: Str(80%) and Ste(20%). The gradient descent algorithm have been performed on the Str and Ste would be used to determine the stopping point presented previously in the subsection 2.3.

Note that the beginning of the overfitting depends on the learning rate chosen in advance and thus computationally the number of passes necessary to find the stopping time. In our case we used the same ones as used at the beginning of this subsection1.

3.3 Comparaison

We present the following table giving the prediction error for each regularization method presented above

Table 2: Performances comparaison

Regularization techniques	Prediction error (MSE) (in Residential Dataset)	Cross entropy (in Forest mapping Dataset)
Ridge regularization	6.29578	0.55506
Averaging SGD (one pass)	6.53099	0.95836
10 passes SGD	6.22346	0.56149
Early stopping	6.07256	0.60666

The regularization capacity of stochastic methods is empirically established through our examples of datasets used. It should be noted in passing that their performance is strongly linked to the chosen learning rate. Although less efficient than the other averaging one pass requires less computational time and complexity (computationally, complexity after one pass $\mathcal{O}(nd)$). However the convergence to an asymptotic region is verified and this is just after a few iterations (4b,4a). Although our results show better performance by doing SGD in several passes, it is nevertheless true that averaging in just one pass is 'sufficient' in the sense that two stochastic techniques met do not give too different results (*linear regression in residential data*). Ridge regularization

gives fairly good performance on our two datasets and is better for logistic regression (*logistic regressionForest Mapping data*). And yet in both datasets.

Note that early stopping gives the best performances in our both datasets and that the method proposed in 2.3 succeeds quite well in finding the beginning of the overfitting.

4 Conclusion

In this project, a comparison between some regularization methods was made in case of least squares regression and logistic regression. Real data were used in this case. Overall, given the data used, early stopping is the most "efficient" regularizer. It should be noted that there are other regularization methods that are not presented in this document. Of course, we would have liked to look at the influence of the learning rate on the SGD methods, to extend the comparison to other methods and to variants (e.g.: theoretical bound in the case of earling stopping,...) of those presented here,... but for the sake of time, we have devoted ourselves only to the most well-known methods.

Appendices

A Ridge complexity

When $d \leq n$, multiplying $\mathbf{X}^\top$ with $\mathbf{X}$ takes $\mathcal{O}(nd^2)$. Inverting $\mathbf{X}^\top\mathbf{X} + \lambda I_d$ takes $\mathcal{O}(d^3)$, multiplying X T with y takes $\mathcal{O}(nd)$ and multiplying $(\mathbf{X}^\top\mathbf{X} + n\lambda I_d)^{-1}$ with $\mathbf{X}^\top\mathbf{Y}$ takes $\mathcal{O}(d^2)$. So in this case, asymptotically $\mathcal{O}(nd^2)$ dominates everything.

B Influence of constant learning rate in SGD

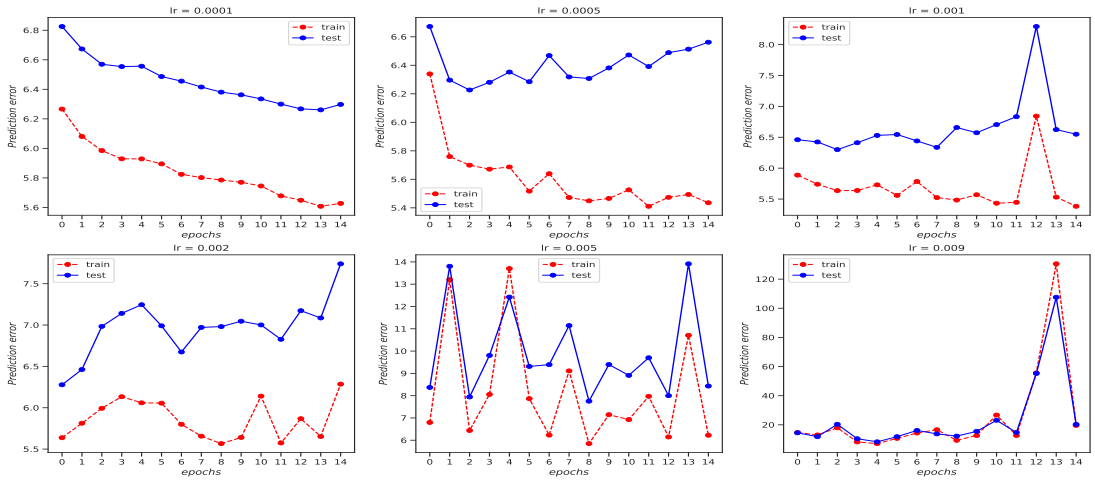


Figure 7: *Influence of constant learning rate in Stochastic Gradient Descent*

References

Boris T. Polyak and Anatoli. B. Juditsky. (1992). Acceleration of stochastic approximation by averaging. *Automation and Remote Control 30.4*, 838–855.

Jastrzębski, S., Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey (2017). Three factors influencing minima in SGD.

Jin, B. and X. Lu (2018). On the Regularizing Property of Stochastic Gradient Descent.

Keskar, N. S., D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang (2017). On large-batch training for deep learning: generalization gap and sharp minima.

Lin, J., R. Camoriano, and L. Rosasco (2016). Generalization properties and Implicit Regularization for Multiples Passes SGM.

Prechelt, L. (2014). *Neural Networks: Tricks of the Trade*, Volume 7700, Chapter Early Stopping - but when?, pp. 53–67.

Raskutti, G., M. J. Wainwright, and B. Yu (2014). Early Stopping and Non-parametric Regression: An Optimal Data-dependent Stopping Rule. *Journal of Machine Learning Research 15*, 365–366.

Robbins, H. and S. Monro (1951). *A stochastic approximation method.The annals of mathematical statistics.*

Tong Zhang (2004). Solving Large Scale Linear Prediction Problems using Stochastic Gradient Algorithms. *ICML '04: Proceedings of the twenty-first international conference on Machine learning*.

Xu, W. (2011). Towards Optimal One Pass Large Scale Learning with Averaged Stochastic Gradient Descent.