Statistics and Data Science

Department of Statistics and Data Science

Computationally Intensive Statistical Methods

Online Learning algorithms for High-dimensional Classification and Regression

Shuo Feng (sf587@cornell.edu)

Jiayin(Joy) Li (jl3724@cornell.edu)

May, 2021

Contents

List of Figures

1 Introduction

The areas of Online Learning Algorithms in Machine Learning is becoming increasingly crucial in prediction and estimation problems.

Online learning is a framework of algorithms which use limited information to build estimated models for future prediction. In contrast to traditional estimation algorithms that use whole data to construct estimation model, online learning algorithms build estimation model by processing one data at a time. Computational efficiency of online learning algorithm is not only significantly obvious in large data processing, but also theoretically guaranteed of high-performance in various scenarios of data. This report will introduce theoretical concepts behind online learning algorithm and present several popular algorithms as well as mathematical driven formula. Online learning algorithms will be implemented in both regression data and classification data. The analysis of estimation and prediction performance includes comparsions of coefficients estimation and online prediction accuracy, theoretical proof and further analysis of proposed algorithms. More specifically, we provide an in-depth description of online mirror descent with mathematical principles and its convergence performance as the tuning of parameters.

In online learning framework, define **loss** function as

$$f_t\beta_t = \beta_t^T X - Y$$

which is the error between estimated y and actual y in tth iterations, with β_t as the estimated β in tth iteration. In the scenario of stochastic learning, since only one data is randomly selected from the sample to compute for β_t, therefore, the loss function becomes **pairwise loss** function for the X_t and Y_t in the tth iteration, defined as

$$f_t\beta_t = \beta_t^T X_i - Y_i$$

Online learning algorithms are implemented sequentially by evaluating the loss function, and pairwise loss function are compared cross periods to observe the T iterations estimation.

Besides the loss function, in online learning, the goal is to minimize the so-called **regret**, which is defined as[Shalev-Shwartz, 2012]:

$$regret(T) = \sum_{t=1}^{T} f_t\beta_t - min \sum_{t=1}^{T} f_t\beta$$

which measures the overall prediction performance of the algorithm compares to the single best decision in hindsight. Online learning algorithms can generate strategy in which the regret is growing sublinearly of the order $O(\sqrt{T})$, for which:

$$\sum_{t=1}^{T} f_t\beta_t - min \sum_{t=1}^{T} f_t\beta \leq O(\sqrt{T})$$

if divide the above term by T on both sides:

$$\frac{1}{T}\sum_{t=1}^{T} f_t\beta_t - min \sum_{t=1}^{T} f_t\beta \leq O(\frac{1}{\sqrt{T}})$$

this is the bound of average regret, which indicates that the average regret approaches to 0 at the rate of $O(\sqrt{T})$ in long run. In particular, the Online Mirror Descent (OMD) algorithm and its is much more powerful because it can achieve regret of order $O(\sqrt{\frac{logd}{T}})$. The powerful performance of OMD, compared with other algorithms, in high-dimensional data will also be discussed in the report.

The rest part of this report will introduce mechanism and iteration algorithms for Recursive Least Square, Online Gradient Descent, Adaptive Gradient Descent, Online Mirror Descent and Composite Objective Mirror Descent; implement those proposed algorithms on both high-dimensional regression problem data and classification problem data; discuss the OMD algorithm on convergence rate and learning rate; summary algorithm results and visualize all estimation and prediction comparison plot.

2 Methodology

2.1 Recursive Least Squares

Recursive Least Squares is used when data are obtained sequentially and want to update estimation with each new data. RLS is to recursively compute the least squares estimate. Specifically, with a known estimate $\hat{\beta_{t-1}}$ after $t-1$ data measurements, and obtain a new pair of data x_t andy_t, RLS can update the estimate to β_t based on previous estimated β and newly-input x_t and y_t. With a linear system:

$$Y = \beta^T X$$

where for each iteration t, the goal is to find the estimate of coefficients β_t that minimize the loss

$$f_t = (y_t - \beta_t^T x_t)$$

equivalently,

$$f_t(\hat{x}) = \frac{1}{2} learning \times \sum_{t=1}^{n} (y_t - \beta^T(t)\hat{x})^2$$

where *learning* is learning rate indicating the weights put on newly input data, usually *learning* can be a constant between 0 and 1, but it can be a learning rate function depending on t, i.e λ_t. Solution to the loss function has a closed form solution as

$$\hat{x} = (\sum_{t=1}^{n} \beta_t \beta_t^T)^{-1} (\sum_{t=1}^{n} \beta_t y_t)$$

Since it is more efficiently to update estimated coefficients as long as a newly data input, therefore, the recursive form is as following:

$$\hat{x_t} = \hat{x_{t-1}} + L_t(y_t - \beta_t^T \hat{x_{t-1}})$$

Letting P_t as the covariance matrix and,

$$L_t = P_{t-1}\beta_t (learning + \beta_t^T P_{t-1} \beta_t)^{-1}$$

$$P_t = (I - L_t \beta_t^T) P_{t-1} \frac{1}{learning}$$

RLS Algorithms to Update Estimate Coefficients.

Algorithm 1 Recursive Least Squares

procedure RLS(x, y)
  $\beta \leftarrow NA$
  $\beta_{initial} \leftarrow 0$
  $V_0 \leftarrow aI$
  for i in 1:n **do**
    $V_{t+1} = V_t - \frac{V_t x_{t+1}^T x_{t+1} V_t}{1 + x_{t+1} V_t x_{t+1}^T}$
    $\beta_{update} = \beta_t + V_{t+1} x_{t+1}^T (y_{t+1} - x_{t+1} \beta_t)$
    store $\beta_i \leftarrow \beta_{update}$
    $\beta_{initial} \leftarrow \beta_{update}$
  return β

2.2 Online Gradient Descent

In the case of extremely large high-dimensional data, RLS will be slow because for each iteration RLS will use the full data set to estimate new β, whereas Online Gradient Descent(OGD)Bubeck1 and Cesa-Bianchi2, 2012 behaviors better than RLS since OGD only use the newly one pair of x_t,y_t to compute estimate of new β. In this strategy, updating coefficients is accelerated. Let η be the learning rate indicating the weights put on newly input data, usually η can be a constant between 0 and 1, but it can be a learning rate function depending on t, i.e ηt.Online gradient updates its β_t vector as follows:

$$\beta_t = \prod_{\omega} (\beta_{t-1} - \eta f_{t-1}) \tag{1}$$

$$= argmin_{\beta \in \omega} \eta f_t(\beta) + \frac{1}{2} ||\beta - \beta_{t-1}||^2 \tag{2}$$

Let $\nabla f_t(\beta_t)$ be the gradient of loss function, and η_t be the learning rate for each iteration, OGD Algorithms to Update Estimate Coefficients.

Algorithm 2 Online Gradient Descent

procedure ODG(x, y, η)
  $\beta \leftarrow NA$
  $\beta_{initial} \leftarrow 0$
  for i in 1:n **do**
    $\beta_{update} = \beta_t - \eta_t \nabla f_t(\beta_t)$
    store $\beta_i \leftarrow \beta_{update}$
    $\beta_{initial} \leftarrow \beta_{update}$
  return β

2.3 Adaptive Gradient Descent

Adaptive Gradient Descent is an improvement to traditional gradient descent algorithms. Contrary to ordinary OGD, Adagrad adapts its learning rate η during its iteration and it updates its parameters β_t separately in each iteration t, η adapts on its own for each β_t

$$\beta_{t+1,i} = \beta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii}} + \epsilon} \nabla_\beta f(\beta_t)$$

where $G_{t+1,i}$ is a matrix containing the squared sum of the past gradients with regards to all β along its diagonal, and ϵ is the generally small correction term.

$$G_t = \sum_{i=1}^{t} \nabla f_i(\beta_i) \nabla f_i(\beta_i)^T$$

The summation of the squared gradients in G_t results the learning rate $\frac{\eta}{\sqrt{G_{t,ii}+\epsilon}}$ to reduce over iteration times. Moreover, it produces low learning rate to frequent occurring features, and high learning rate to infrequent features. A closed form of β udpate in Adagrad is :

$$\beta_{t+1} = \beta_t - \eta diag(G_t)^{-\frac{1}{2}} \nabla f_t(\beta_t)$$

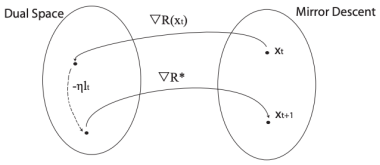
Adagrad Algorithms to Update Estimate Coefficients.

Algorithm 3 Adaptive Gradient Descent

procedure ADAGRAD(x, y, η)
  $\beta \leftarrow NA$
  $\beta_{initial} \leftarrow 0$
  for i in 1:n **do**
    $G_t = \sum_{i=1}^{t} \nabla f_i(\beta_i) \nabla f_i(\beta_i)^T$
    $\beta_{update} = \beta_t - \eta diag(G_t)^{-\frac{1}{2}} \nabla f_t(\beta_t)$
    store $\beta_i \leftarrow \beta_{update}$
    $\beta_{initial} \leftarrow \beta_{update}$
  return β

2.4 Online Mirror Descent

Among online learning algorithms, a unifying perspective on the design and the analysis of online algorithms is provided by online mirror descent. Online Mirror Descent is a first-order optimization procedure which generalizes the classic Gradient Descent procedure to non-Euclidean geometries by relying on a "distance generating function" named as Bregman divergences,a strictly convex function which measures the distance between two points. Online Mirror Descent(OMD)Kunapuli and Shavlik, 2012 is an iterative method for minimizing a convex function $\psi_t(\beta_t) : \omega \in R$. Additionally,its convergence and effectiveness guarantees can be proved theoretically by instantiating the general OMD bounds to the specific regularizer being used.



A main concept in the online mirror descent is the notion of strong convexity:
Define $\psi : B \to \mathbb{R}$ is a convex function $w.r.t$ |||| if for any $\beta, \beta^{'} \in B$:

$$\forall_{\alpha\in[0,1]}\psi(\alpha\beta + (1-\alpha)\beta^{'}) \leq \alpha\psi(\beta) + (1-\alpha)\psi(\beta^{'}) - \frac{\alpha(a-\alpha)}{q}||\beta - \beta^{'}||^{q}$$

Since norm ——·—— and subset β is not related, therefore only require convexity insideβ.
The mirror map function ψ is often chosen to be differentiable and strongly convex $w.r.t$ some norm ——·—— with modulus q, that is,

$$\psi(x) \geq \psi(y) + < \vec{\boldsymbol{\nabla}}\psi(y), x - y > + \frac{\rho}{2}||x-y||^{2}$$

Then, associate Bregman Divergence,generalized distance measure, with ψ,defined as

$$B_{\psi}(x||y) = \psi(x) - \psi(y) - \nabla\psi(y)^{T}(x-y)$$

Proof of Bregman divergence and convex conjugacy:
As above defined, ψ be a differentiable and strictly convex function on C, and $\psi^{\star}$ is also differentiable and strictly convex on $\nabla\psi(C)$, then consider the following Bregman divergence:

$$B_{\psi^{\star}}(u||v) = \psi^{\star}(u) - \psi^{\star}(v) - \nabla\psi^{\star}(v)^{T}(u-v)$$

Consider a pair $x, z \in C$ such that $u = \nabla\psi(x)$, and $v = \nabla\psi(z)$. Since (x, u) and z, v are dual pairs, which indicates $\psi^{\star}(u) + \psi(x) =< x, u >$, same as for (z, v), therefore,

$$B_{\psi^{\star}}(u||v) = \psi(z) - \psi(x) - \nabla\psi(x)^{T}(z-x)$$

Thus, conjugacy proved as

$$B_{\psi^{\star}}(\nabla\psi(x)||\nabla\psi(z)) = B_{\psi}(z||x)$$

If choose $\psi(x) = \frac{1}{2}||x||_{2}^{2}$, then $B_{\psi}(x||y)$ is the Euclidean distance, and the algorithm is OGD.

When the dimension d is large enough, OMD is optimal among first-order methods because each iterations minimizes the first order Taylor expansion of the loss function f_t with a regularization term specified by the Bregman divergence. It is possible to bound the regret as

$$\sum_{t=1}^{T}\psi_t(\beta_t) - inf_{\beta\in\omega}\sum_{t=1}^{T}\psi_t(\beta)$$

in which the guarantee on online regret can be translated directly to a guarantee on the convergence rate of the algorithm will shown later.

Following Beck and Teboulle's exposition, a Mirror Descent update in the online setting can be formed as

$$\beta_{t+1} = argmin_{\beta\in\omega} f_t(\beta_t) + \nabla f_t(\beta_t)^T(\beta - \beta_t) + \frac{1}{\eta} B_\psi(\beta||\beta_t)$$

$\nabla f_t(\beta_t)$denotes subgradient of $f_t(\beta_t)$. Intuitively, OMD works by minimizing a first-order approximation of the function β_t at tth iteration and force the next iterate β_{t+1} to lie close to β_t. The learning η controls the trade-off.

Choose appropriate mirror map function and associated Bregman divergence is important for OMD. According to some references, choose $\psi(x)$ as the l_p-norms squared with p theoretically selected based on the dimension d.

$$\psi(x) = \frac{1}{2}||x||_p^2$$

Then is the step to choose p. Choose $p = 1 + \frac{1}{logd}$, matching Lipschitz assumption ($1 \leq p \leq 2, \frac{1}{q} + \frac{1}{p} = 1$)was shown to be optimal in terms of the number of exact gradient evaluations, with $\psi(x) = \frac{1}{2}||x||_p^2$ being used.

The last setup step is to find the inverse of $\nabla\psi(x)$ which helps in the closed form solution of estimate β_t update. With $\psi(x) = \frac{1}{2}||x||_p^2$, the inverse of $\nabla\psi(x)$ has a solution as[Gentile, 2003]:

$$(\nabla\psi(x))^{-1} = \frac{sign(x)|x|^{q-1}}{||x||_q^{q-2}}$$

Finally, with all above selection, generate a closed form solution of estimate β_t update by taking derivative equals 0 of the following $argmin$ function.

By decomposition, OMD algorithm is equivalently to solve the following for $t = 1, 2, ..., T$:

$$\nabla\psi(\beta_{t+\frac{1}{2}}) = \nabla\psi(\beta_t) - \eta\nabla f_t\beta_t \tag{3}$$

$$\beta_{t+1} = argmin_{\beta\in\omega} B_\psi(\beta||\beta_{t+\frac{1}{2}}) \tag{4}$$

Solving the $argmin$ function, get the closed form solution:

$$\nabla\psi(\beta_{t+1}) = \nabla\psi(\beta_t) - \eta\nabla f_t(\beta_{t+1})$$

OMD Algorithms to Update Estimate Coefficients.

2.5 Composite Objective Mirror Descent

In the Composite Objective Mirror Descent (COMID)[Duchi et al., 2010], the goal is to generalize mirror descent for the case when the functions ψ_t are composite, that is, the loss funcition for each

Algorithm 4 Online Mirror Descent

procedure OMD(x, y, η)

$p \leftarrow 1 + \frac{1}{logd}$

$q \leftarrow \frac{p}{p-1}$

$\beta \leftarrow NA$

$\beta_{initial} \leftarrow 0$

for i in 1:n **do**

$wt = \nabla\psi(\beta_t)$

$gt = \eta\nabla f_t(\beta_t)$

$zt = wt + gt$

$\beta_{update} = \frac{sign(zt)|zt|^{q-1}}{||zt||_q^{q-2}}$

store $\beta_i \leftarrow \beta_{update}$

$\beta_{initial} \leftarrow \beta_{update}$

return β

iteration is:$\psi_t = f_t + r$, with the modificaiton of loss function, the β_t estimate update is:

$$\beta_{t+1} = argmin_{\beta\in\omega}\eta < f_t'(\beta_t, \beta) > + B_\psi(\beta, \beta_t) + \eta r(\beta)$$

This is nearly the same structure as OMD update only with difference of no linearization of r, this is why so called Composite Objective Mirror Descent. In many situations, the COMID update is no costlier than the usual OMD update. In these situations, each COMID update is efficient and benefits from the presence of the regularizer $r(\beta)$. COMID has a regret bounds $O(\sqrt{logT})$ when composite functions $f_t + r$ are strongly convex, also, with various settings of Bregman function ψ and regularizer r such as ridge regression, Lasso regression and support vector machine, etc., performance of COMID is different.

For regularized online learning, the goal is to achieve low regret with a sequence of function $f_t(\beta) + r(\beta)$. In each iteration, generate an estimate $\beta_t \in R$, get the function f_t. The bound of regularized regret against a comparator $\beta^(\star)$ is:

$$R_{f_t+r}(T) = \sum_{t=1}^{T}(f_t(\beta_t)) + r(\beta_t) - f_t(w^\star) - r(w^\star)$$

Remind that from above, ψ function designates a continuously differentiable function that is $\alpha-$strongly convex w.r.t a norm |||| on the set ω, and $B_\psi(x, y)$ satisfies that:

$$B_\psi(x, y) \geq \frac{\alpha}{2}||x - y||^2$$

Then can proved that:

$$R_\psi(T) \leq \frac{1}{\eta}R_\psi(\beta^\star + \beta_0) + \frac{\eta}{2\alpha}\sum_{t=1}^{T}||f_t'(\beta_t)||_\star^2$$

COMID Algorithms to Update Estimate Coefficients.

Algorithm 5 Composite Objective Mirror Descent

procedure COMID(x, y, r, η)

$p \leftarrow 1 + \frac{1}{logd}$

$q \leftarrow \frac{p}{p-1}$

$\beta \leftarrow NA$

$\beta_{initial} \leftarrow 0$

for i in 1:n **do**

$\beta_{update} = argmin_{\beta \in \omega} f_t(\beta_t) + r(\beta_t) + f_t(\beta_t) +_t (\beta_t)^T(\beta - \beta_t) + \frac{1}{\eta} B_\psi(\beta||\beta_t)$

store $\beta_i \leftarrow \beta_{update}$

$\beta_{initial} \leftarrow \beta_{update}$

return β

2.6 Polyak-Ruppert Averaging

Averaged stochastic gradient descent, invented independently by Ruppert and Polyak in the late 1980s, is ordinary stochastic gradient descent that records an average of its parameter vector over time. That is, the update is the same as for ordinary stochastic gradient descent, but the algorithm also keeps track of Polyak and Juditsky, 1992 Ruppert, 1988

$$\bar{\beta}_t = \frac{1}{t} \sum_{i=1}^{t} \beta_i$$

When optimization is done, this averaged parameter vector takes the place of β.

3 High Dimensional Data Analysis

In this project we will investigate effectiveness of different methods mentioned in Section 2 through simulation data. In particular, we will explore how the prediction and parameter estimation performances of these algorithms change. We will also explore strategies for minimizing logistic loss functions specific to classification problems.

3.1 Regression

For linear regression, we consider i.i.d $x_t \sim N(0, I_t)$ and $y_t = x_t^T \beta + \epsilon_t$, where $\epsilon_t \sim N(0, 1)$ and β is a sparse vector. Our simulation design matrix has $n = 1000$ samples, and dimension $d = 2000$. For sparse vector β, the first 200 entries are drawn from i.i.d. $\sim N(0, 5^2)$, and the rest 1800 entries are zero. With linear regression framework, we try to minimize the l_2 loss function, i.e. for each step of iteration, we predict $\hat{y}_t = x_t^T \beta_t$, and incur loss $(y_t - x_t^T \beta_t)^2$. We calculate and compare mean square prediction error(MSPE) and estimation error $\|\beta - \beta_t\|_2$.

Figure 1 shows the coefficients estimation of each algorithm. We notice that RLS and AdaGrad give similar performance results, they fail to distinguish the contribution of signal variables and noise variables, and they give similar estimation for zero and non-zero coefficients. OGD has a better performance for non-zero coefficients but still gives too large estimates for zero coefficients.

However, the OMD algorithm gives remarkably more accurate coefficient estimates, it quite successfully distinguishes the true signal variables from the noisy variables by giving the former estimated values in noticeably larger magnitude than the latter.

In Figure 2 and Figure 3, we further explore how the prediction and parameter estimation performances of these algorithms change as the number of predictors grows ($n = 8000$). We notice that OMD outperforms other algorithms for small sample size n, while when n is gradually getting larger, these algorithms have close prediction errors and estimation errors. We investigate the performance of AdaGrad with different step size. Figure 4 gives the prediction error of AdaGrad with step sizes $0.1, 0.5, 1, 2$, it demonstrates that like standard gradient, AdaGrad is also sensitive to step-size selection.

We are interested in the effectiveness of these algorithms on real data set, the real dataset used in this example is the riboflavin dataset, which contains measures of gene expression on $n = 71$ sample observations of $p = 4088$ predictor genes. The response variable in this case is continuous, and represents the log-transformed riboflavin production rate. This dataset is contained in the `hdi` R package Dezeure et al., 2015. Due to the fact that the size of the riboflavin dataset is small, we choose to allocate a higher proportion to the training set: the training ($n_1 = 60$) and test set ($n_1 = 11$) respectively. Table 1 shows the testing error for different methods, we notice that real High dimensional real data set, method of dimension reduction (AdaGrad) significantly outperform the classical learning algorithms (RLS, OGD, OMD).

RLS	OGD	AdaGrad	OMD
NaN	1.766607e+267	2.893	48.404

Table 1: Testing error for riboflavin dataset

3.2 Classification

Under logistic regression setting for classification, we consider the same design matrix as linear regression, i.e. i.i.d $x_t \sim N(0, I_t)$ and binary response variable $y_t \in \{0, 1\}$ is drawn from Bernoulli distribution with $\mathbb{P}(y_t = 1|x_t) = \frac{1}{1+\exp(x_t^T \beta_t)}$. The true coefficients β is still the sparse vector, with the first 200 entries are drawn from i.i.d. $\sim N(0, 5^2)$, and the rest 1800 entries are zero. We try to minimize the cross-entropy loss function, i.e. for each step of iteration, we predict $\hat{y}_t = \frac{1}{1+\exp(x_t^T \beta_t)}$, and incur loss $-[y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$, and we still calculate and compare mean square prediction error(MSPE). We exclude the RLS here because it cannot be adapted to the logistic regression setting, and we use COMID instead.

Figure 5 shows the estimation of the coefficients of each algorithm, and figure 6 shows the prediction error for classification. We notice that none of the methods performs very well in terms of coefficient estimation. OGD and AdaGrad fail to distinguish the contribution of signal variables and noise variables, and they give similar estimations for zero and non-zero coefficients. Although OMD and COMID have a better performance for non-zero coefficients it still gives too large estimates for zero coefficients. OMD and COMID have slightly smaller prediction errors with small sample size, but when n is gradually getting larger, these algorithms have close prediction errors. In Figure

7, we apply Polyak-Ruppert Averaging to OGD, we realize that although the Averaging doesn't significantly improve the prediction accuracy, it gives a smooth prediction error instead of wiggly line, so it reduce the risk of any abrupt rise or descend in prediction error.

We further investigate some properties of OMD and COMID. We are interested in the convergence rate of OMD, Duchi et al., 2010 proposed a theorem of the convergence rate.
Theorem: Suppose $\boldsymbol{\beta} = 0$. Let $p = 1 + 1/\log(d)$ and use the $\psi(x) = \frac{1}{2}||x||_p^2$. Further suppose that either Ω is compact or the f_t are Lipschitz so that for $q = \log d + 1$, $\max_t \|f_t'(\beta_t)\|_q \leq G_q$.Setting $\eta = \frac{\|\beta^*\|}{G_q}\sqrt{1T\log d}$, the regret of OMD satisfies

$$\frac{1}{T}\sum_{t=1}^{T} f_t(\beta_t) - \frac{1}{T}\sum_{t=1}^{T} f_t(\beta) \leq \|\beta\|_p G_q \sqrt{\frac{\log d}{T}}$$

We notice that for logistic regression, we are able to bound $\|f_t'(\beta_t)\|_q$, since $\|f_t'(\beta_t)\|_q = \|x_t\|_q$. Hence, we generate a new design matrix with bounded x_t i.i.d unif$(-3, 3)$, and with different dimension $d = 500, 1000, 2000, 5000$, and figure 8 shows the prediction error versus the sample size, and four lines with different dimensions are deviate from each other. This validates that the prediction error depends on $\log(d)$. However, if we consider the theoretical learning rate $\mathcal{O}(\sqrt{logd/T})$ the empirical convergence rate of prediction error for the OMD is much slower than than $\mathcal{O}(\sqrt{logd/T})$.

For COMID, we investigate how tuning parameter affect the prediction error, we choose three different tuning parameters $\lambda = 0.05, 0.5, 1$. Figure 9 shows the comparison the prediction error for different tuning parameters is more or less overlapping, indicating that they have close accuracy. While for stronger regularization, the prediction error is getting extremely large, so with mild regularization, the regularization in COMID didn't make a big difference to the results.

References

Bubeck1, S ebastien and Nicolo‘ Cesa-Bianchi2 (2012). ‘Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems’. In:

Dezeure, Ruben et al. (2015). ‘High-dimensional inference: Confidence intervals, p-values and R-software hdi’. In: *Statistical Science* 30(4), pp. 533–558.

Duchi, John et al. (2010). ‘Composite Objective Mirror Descent’. In: *The 23rd Conference on Learning Theory*.

Gentile, C. (2003). ‘The robustness of the p-norm algorithms.’ In: *Machine Learning* 53(3), pp. 265–299.

Kunapuli, Gautam and Jude Shavlik (2012). ‘Mirror Descent for Metric Learning: A Unified Approach’. In: *Proc. 23rd European Conf. on Machine Learning*.

Polyak, B. T. and A. B. Juditsky (1992). ‘Acceleration of stochastic approximation by averaging’. In: *SIAM Journal on Control and Optimization* 30(4), pp. 838–855.

Ruppert, D. (1988). ‘Efficient estimations from a slowly convergent robbins-monro process’. In: *Technical Report 781, Cornell University Operations Research and Industrial Engineering*.

Shalev-Shwartz, S. (2012). ‘Online learning and online convex optimization’. In: *Foundations and Trends in Machine Learning* 4(2), pp. 107–194.

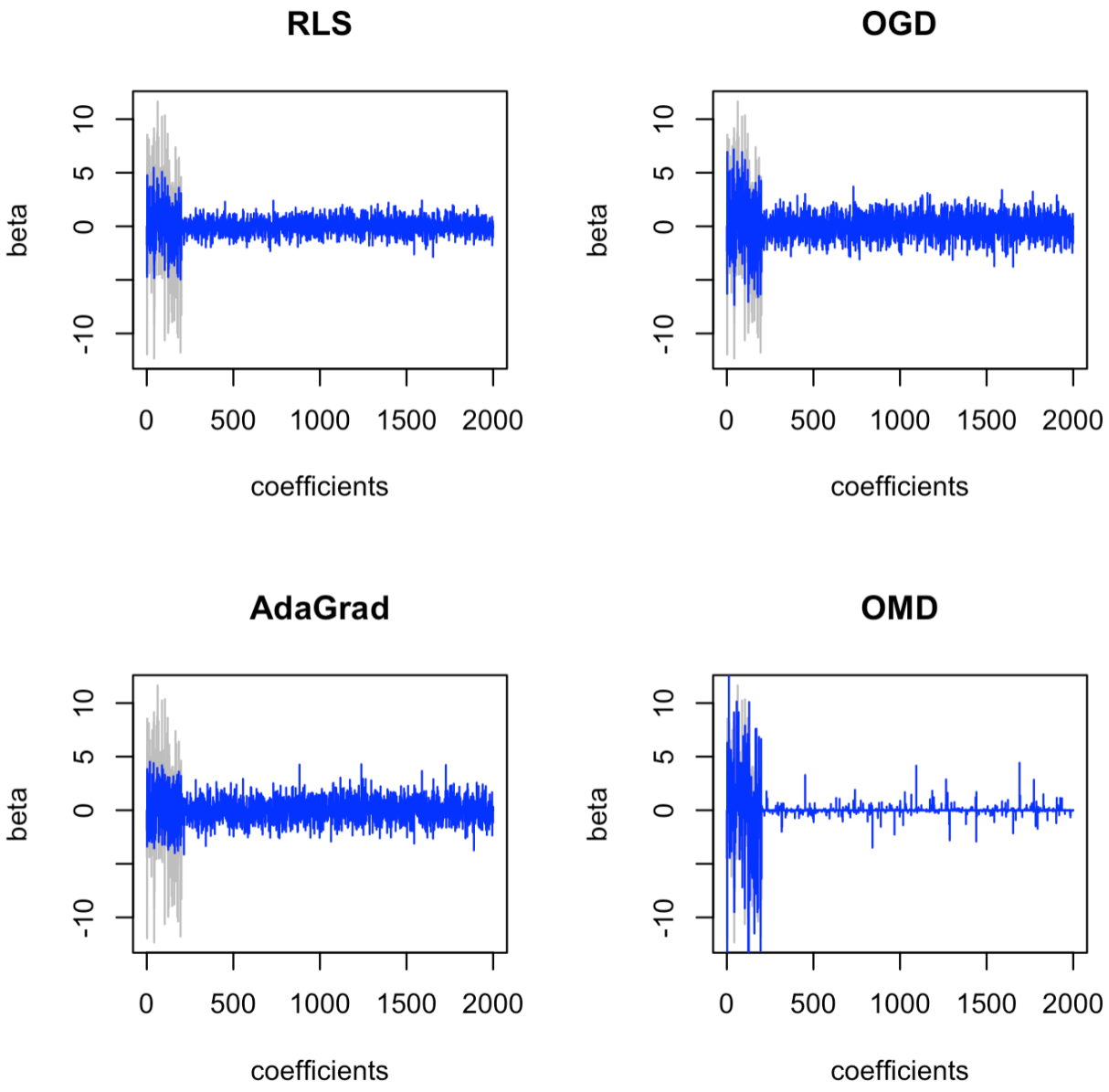


Figure 1: Regression: Comparison of True coefficients (grey) and estimated coefficients (blue) at the last iteration (T=1000, d = 2000)

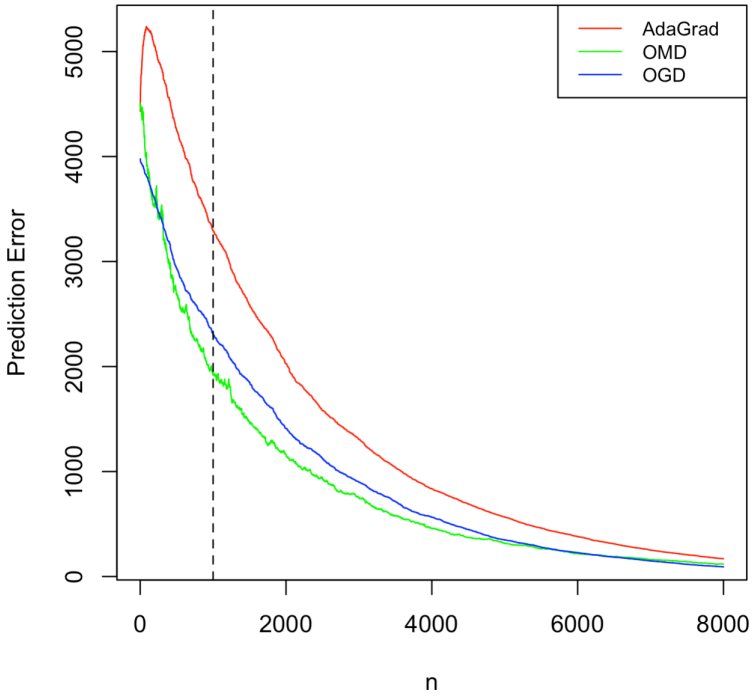


Figure 2: Prediction error of OGD, AdaGrad and OMD for linear regression

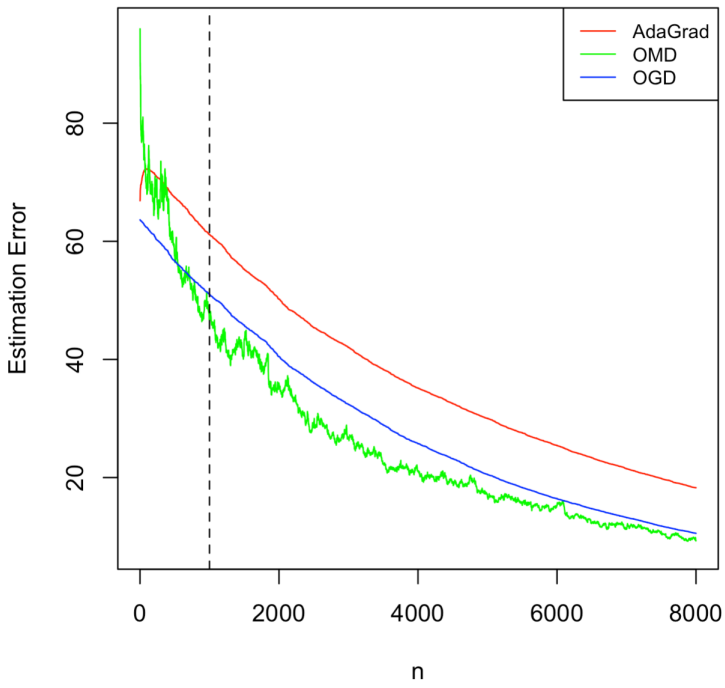


Figure 3: Estimate error of OGD, AdaGrad and OMD for linear regression

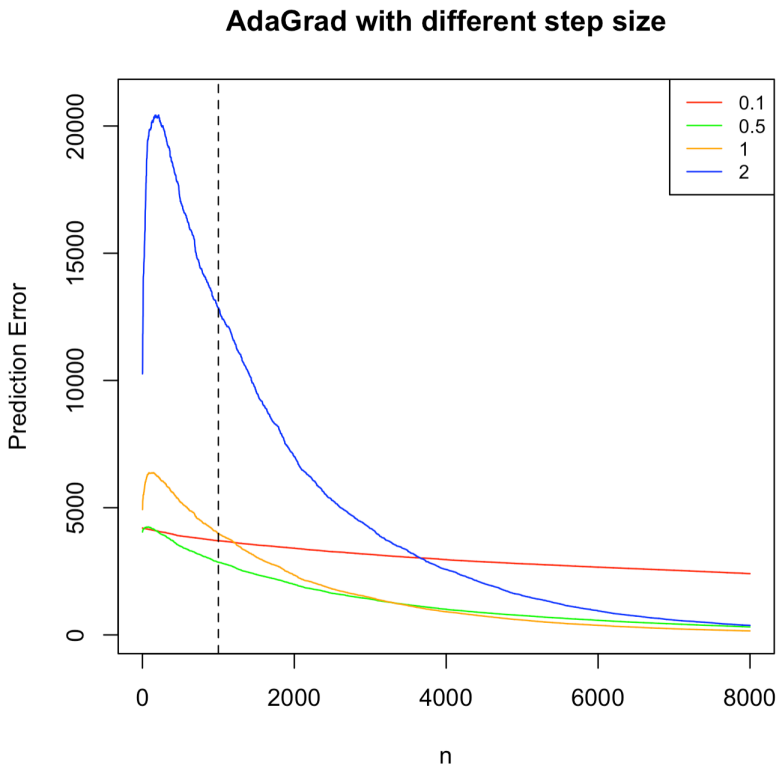


Figure 4: AdaGrad with different step size

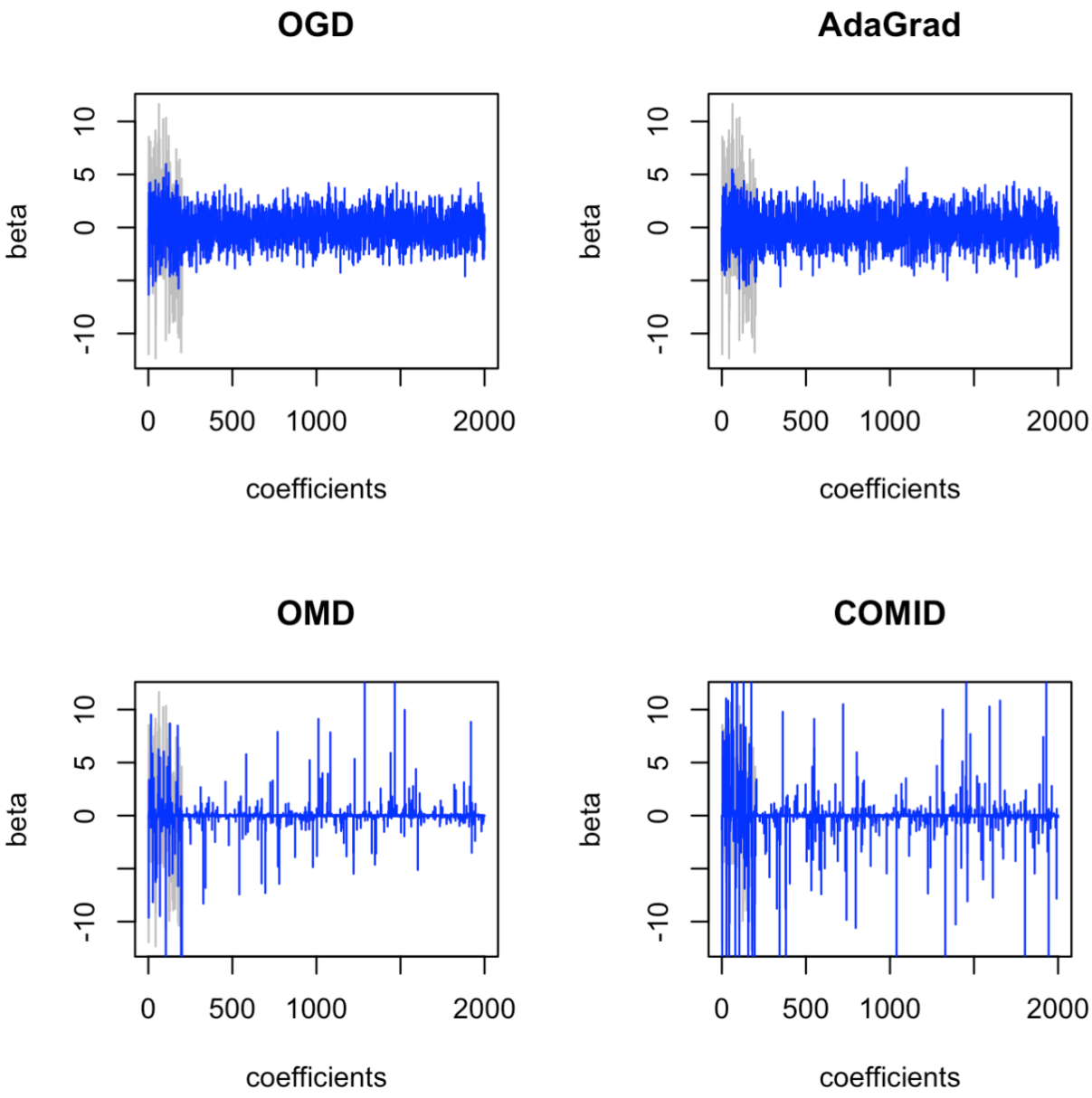


Figure 5: Classification:Comparison of True coefficients (grey) and estimated coefficients (blue) at the last iteration (T=1000, d = 2000)

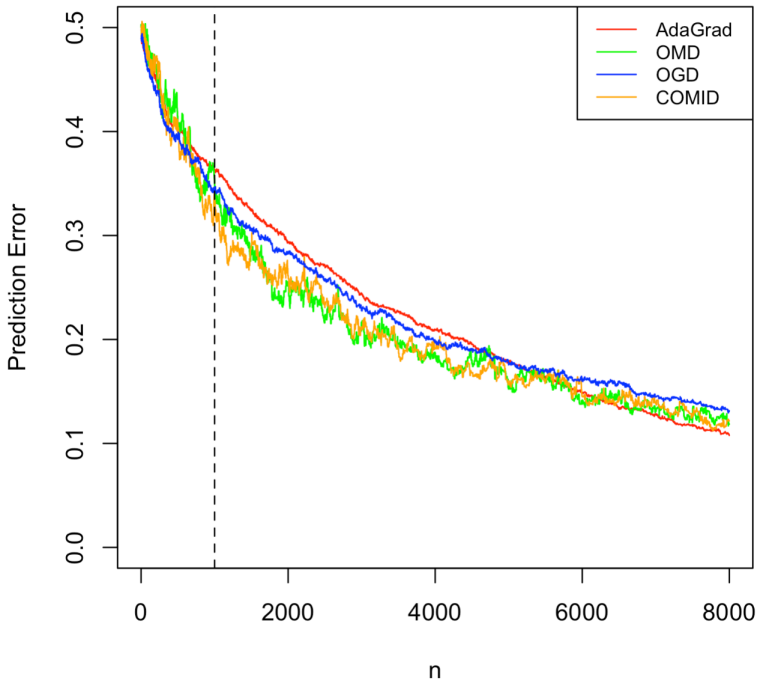


Figure 6: Prediction error of OGD, AdaGrad, OMD and COMID for classification

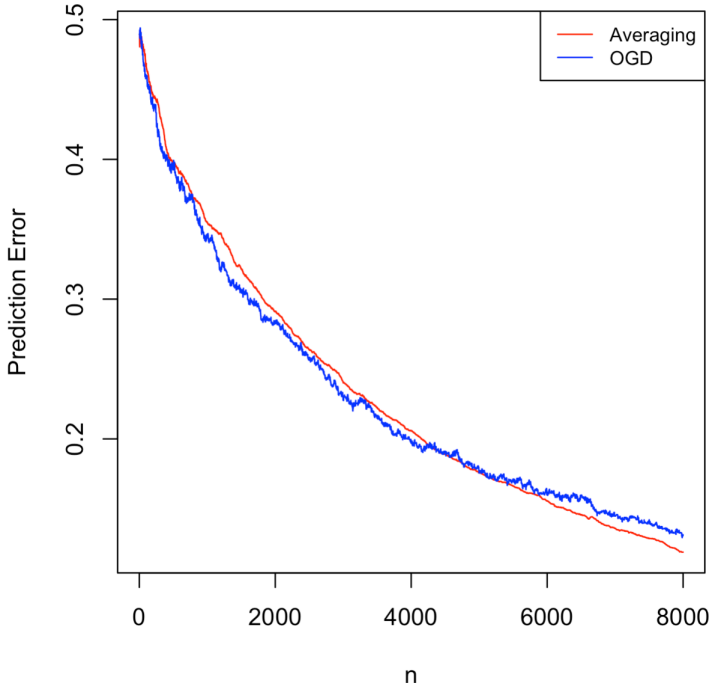


Figure 7: Averaging

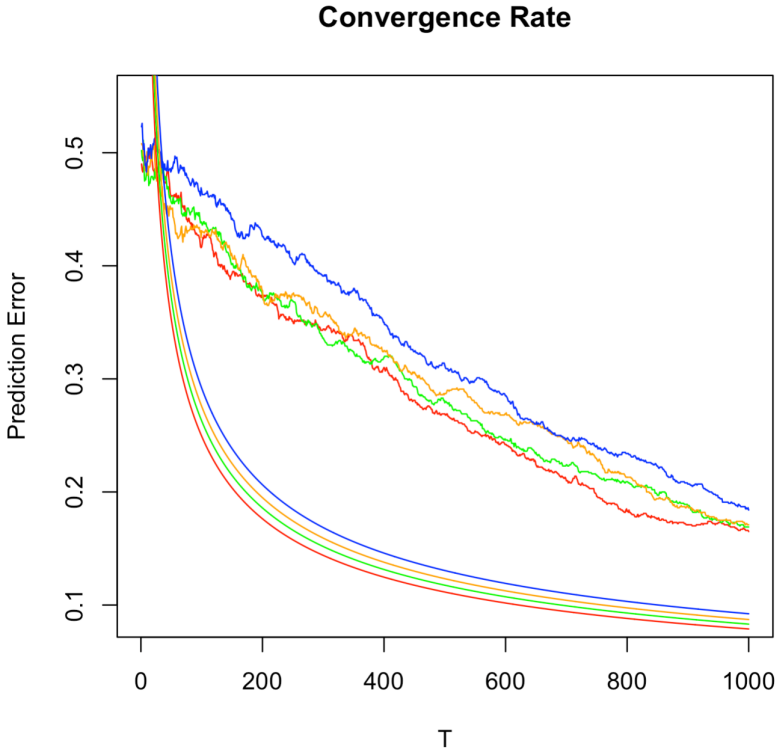


Figure 8: Convergence rate

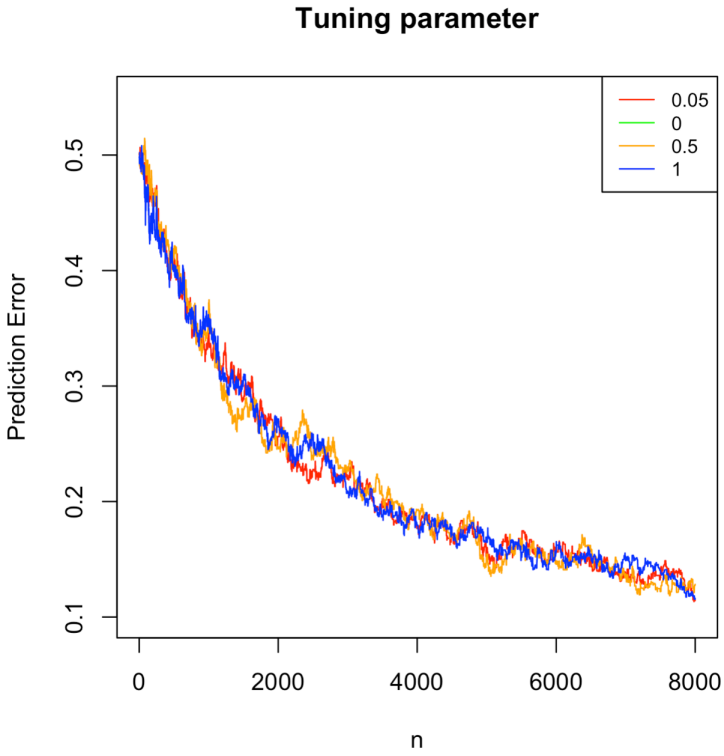


Figure 9: Tuning parameters