Are You Still Tuning Hyperparameters? Parameter-free Model Selection and Learning

Francesco Orabona
Yahoo! Labs
New York, USA
francesco@orabona.com

Abstract

Stochastic gradient descent algorithms for training linear and kernel predictors are gaining more and more importance, thanks to their scalability. While various methods have been proposed to speed up their convergence, the model selection phase is often ignored. In this paper, we propose a new kernel-based stochastic gradient descent algorithm that performs model selection while training, with no parameters to tune, nor any form of cross-validation. The algorithm estimates over time the right regularization in a data-dependent way. Optimal rates of convergence are proved under standard smoothness assumptions on the target function.

1 Introduction

Stochastic Gradient Descent (SGD) algorithms are gaining more and more importance in the Machine Learning community as efficient and scalable machine learning tools. There are two possible ways to use a SGD algorithm: to optimize a batch objective function, e.g. [14], or to directly optimize the generalization performance of a learning algorithm, in a stochastic approximation way [12]. The second use is the one we will consider in this paper. It allows learning over streams of data, coming Independent and Identically Distributed (IID) from a stochastic source.

Both in theory and in practice, the convergence rate of SGD for any finite training set critically depends on the step sizes used during training. In fact, often theoretical analyses assume the use of optimal step sizes, rarely known in reality, and in practical applications wrong step sizes can result in arbitrary bad performance. While in finite dimensional hypothesis spaces simple optimal strategies are known [1], in infinite dimensional spaces the only attempts to solve this problem achieve convergence only in the realizable case, e.g. [16], or assume prior knowledge of intrinsic (and unknown) characteristic of the problem [15, 18, 19, 20]. The only known practical and theoretical way to achieve optimal rates in infinite Reproducing Kernel Hilbert Space (RKHS) is to use some form of cross-validation to select the step size that corresponds to a form of model selection [17]. However, cross-validation techniques would result in a slower training procedure partially neglecting the advantage of the stochastic training. Also, the situation is exactly the same in the batch setting where the regularization takes the role of the step size. Even in this case, optimal rates can be achieved only when the regularization is chosen in a problem dependent way [5, 17].

In this paper we present a novel stochastic parameter-free algorithm, called Parameter-free STOchastic Learning (PiSTOL), that obtains optimal finite sample convergence bounds in infinite dimensional RKHSs. This new algorithm has the same complexity as the plain stochastic gradient descent procedure and implicitly achieves the model selection while training, with no parameters to tune nor the need for cross-validation. The core idea is to change the step sizes over time in a data-dependent way. This is the first algorithm of this kind to have provable optimal convergence rates.

2 Problem Setting and Definitions

Let $\mathcal{X} \subset \mathbb{R}^d$ a compact set and $\mathcal{H}_K$ the RKHS associated to a Mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ implementing the inner product $\langle \cdot\,,\, \cdot \rangle_K$. The inner product is defined so that it satisfies the reproducing property, $\langle K(\boldsymbol{x}, \cdot)\,,\, f(\cdot) \rangle_K = f(\boldsymbol{x})$. For simplicity, we focus on the classification setting and the performance is measured w.r.t. a loss function $\ell : \mathbb{R} \to \mathbb{R}_+$. We will consider *$L$-Lipschitz*, that is $|\ell(x) - \ell(x')| \leq L|x - x'|$, $\forall x, x' \in \mathbb{R}$, and *$H$-smooth* losses, that is differentiable losses with the first derivative H-Lipschitz.

Let ρ a fixed but unknown distribution on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y} = \{-1, 1\}$. A training set $\{\boldsymbol{x}_t, y_t\}_{t=1}^T$ will consist of samples drawn IID from ρ. Denote by $\rho_{\mathcal{X}}$ the marginal probability measure on $\mathcal{X}$ and let $\mathcal{L}^2_{\rho_{\mathcal{X}}}$ be the space of square integrable functions with respect to $\rho_{\mathcal{X}}$. We will assume that the support of $\rho_{\mathcal{X}}$ is $\mathcal{X}$. Define the *ℓ-risk* of f, as $\mathcal{E}^\ell(f) := \int_{\mathcal{X}\times\mathcal{Y}} \ell(yf(x))d\rho$. Also, define $f^\ell_\rho(x) := \arg\min_{t\in\mathbb{R}} \int_{\mathcal{Y}} \ell(yt)d\rho(y|x)$, that gives the *optimal ℓ-risk*, $\mathcal{E}^\ell(f^\ell_\rho) = \inf_{f\in\mathcal{L}^2_{\rho_{\mathcal{X}}}} \mathcal{E}^\ell(f)$. Note that $f^\ell_\rho \in \mathcal{L}^2_{\rho_{\mathcal{X}}}$, but it could not be in $\mathcal{H}_K$ still in some cases it possible to achieve its performance. Now we will introduce a parametrization to consider in a smooth way the case that f^ℓ_ρ belongs or not to $\mathcal{H}_K$.

Let $L_K : \mathcal{L}^2_{\rho_{\mathcal{X}}} \to \mathcal{L}^2_{\rho_{\mathcal{X}}}$ the integral operator defined by $(L_K f)(x) = \int_{\mathcal{X}} K(x, x')f(x')d\rho_{\mathcal{X}}(x')$. There exists an orthonormal basis $\{\Phi_1, \Phi_2, \cdots\}$ of $\mathcal{L}^2_{\rho_{\mathcal{X}}}$ consisting of eigenfunctions of L_K with corresponding non-negative eigenvalues $\{\lambda_1, \lambda_2, \cdots\}$ and the set $\{\lambda_i\}$ is finite or $\lambda_k \to 0$ when $k \to \infty$ [6, Theorem 4.7]. Since K is a Mercer kernel, L_K is compact and positive. Therefore, the fractional power operator L^β_K is well defined for any $\beta \geq 0$. We indicate its range space by

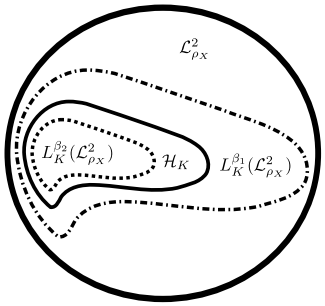

Figure 1: $\mathcal{L}^2_{\rho_{\mathcal{X}}}$, $\mathcal{H}_K$, and $L^\beta_K(\mathcal{L}^2_{\rho_{\mathcal{X}}})$ spaces, with $0 < \beta_1 < \frac{1}{2} < \beta_2$.

$$L^\beta_K(\mathcal{L}^2_{\rho_{\mathcal{X}}}) := \left\{ f = \sum_{i=1}^{\infty} a_i \Phi_i \;:\; \sum_{i:a_i\neq 0} a_i^2 \lambda_i^{-2\beta} < \infty \right\}. \quad (1)$$

By the Mercer's theorem, we have that $L^{\frac{1}{2}}_K(\mathcal{L}^2_{\rho_{\mathcal{X}}}) = \mathcal{H}_K$, that is every function $f \in \mathcal{H}_K$ can be written as $L^{\frac{1}{2}}_K g$ for some $g \in \mathcal{L}^2_{\rho_{\mathcal{X}}}$. On the other hand, by definition of the orthonormal basis, $L^0_K(\mathcal{L}^2_{\rho_{\mathcal{X}}}) = \mathcal{L}^2_{\rho_{\mathcal{X}}}$. Thus, the smaller β is, the bigger this space of the functions will be, see Fig. 1. This space has a key role in our analysis. In particular, we will assume that $f^\ell_\rho \in L^\beta_K(\mathcal{L}^2_{\rho_{\mathcal{X}}})$ for $\beta > 0$, that is

$$\exists g \in \mathcal{L}^2_{\rho_{\mathcal{X}}} \;:\; f^\ell_\rho = L^\beta_K(g). \quad (2)$$

3 PiSTOL: Parameter-free STOchastic Learning

The PiSTOL algorithm is in Algorithm 1. The algorithm builds on recent advancement in unconstrained online learning [8, 9]. It is very similar to an Averaged Stochastic Gradient Descent (ASGD) algorithm [21], the main difference being the computation of the solution based on the past gradients, in line 4. Note that the calculation of $\|g_t\|^2_K$ can be done incrementally, hence, the computational complexity is the same as ASGD, in both finite and infinite dimensional spaces. For the PiSTOL algorithm we have the following convergence guarantee.[1]

Theorem 1. *Assume that the samples $(\boldsymbol{x}_t, y_t)_{t=1}^T$ are IID from ρ, (2) holds for $\beta \leq \frac{1}{2}$. Also, assume that the sequence of $\boldsymbol{x}_t$ satisfies $\|k(\boldsymbol{x}_t, \cdot)\|_K \leq 1$ and the loss ℓ is convex, L-Lipschitz, and H-smooth. Then, setting $a_t = 3L$ and $b_t = 3L$, the solution of PiSTOL satisfies*[2]

- *If $\beta \leq \frac{1}{3}$ then* $\mathbb{E}[\mathcal{E}^\ell(\bar{f}_T)] - \mathcal{E}^\ell(f^\ell_\rho) \leq \tilde{\mathcal{O}}\left(\max\left\{(\mathcal{E}^\ell(f^\ell_\rho) + 1/T)^{\frac{\beta}{2\beta+1}} T^{-\frac{2\beta}{2\beta+1}}, T^{-\frac{2\beta}{\beta+1}}\right\}\right)$.

[1] The proofs are in [10].

[2] For brevity, the $\tilde{\mathcal{O}}$ notation hides polylogarithmic terms.

Algorithm 1 PiSTOL: Parameter-free STOchastic Learning.

Parameters: $a_t, b_t > 0$
Initialize: $g_0 = \mathbf{0} \in \mathcal{H}_K$
for $t = 1, 2, \dots$ **do**
  Set $\alpha_{t-1} = a_t \left(a_t + \sum_{i=1}^{t-1} |\ell'(y_i f_i(\boldsymbol{x}_i))| \, \|k(\boldsymbol{x}_i, \cdot)\|_K \right)$
  Set $f_t = g_{t-1} \frac{b_{t-1}}{\alpha_{t-1}} \exp\left(\frac{\|g_{t-1}\|_K^2}{2\alpha_{t-1}} \right)$
  Receive input vector $\boldsymbol{x}_t \in \mathcal{X}$
  Update $g_t = g_{t-1} - y_t \ell'(y_t f_t(\boldsymbol{x}_t)) k(\boldsymbol{x}_t, \cdot)$
end for
Return $\bar{f}_T = \frac{1}{T} \sum_{t=1}^{T} f_t$

- *If* $\frac{1}{3} < \beta \leq \frac{1}{2}$, *then* $\mathbb{E}[\mathcal{E}^\ell(\bar{f}_T)] - \mathcal{E}^\ell(f_\rho^\ell)$

$$\leq \tilde{\mathcal{O}} \left(\max \left\{ (\mathcal{E}^\ell(f_\rho^\ell) + 1/T)^{\frac{\beta}{2\beta+1}} T^{-\frac{2\beta}{2\beta+1}}, (\mathcal{E}^\ell(f_\rho^\ell) + 1/T)^{\frac{3\beta-1}{4\beta}} T^{-\frac{1}{2}}, T^{-\frac{2\beta}{\beta+1}} \right\} \right).$$

This theorem guarantees consistency w.r.t. the ℓ-risk. We have that the rate of convergence to the optimal ℓ-risk is $\tilde{\mathcal{O}}(T^{-\frac{3\beta}{2\beta+1}})$, if $\mathcal{E}^\ell(f_\rho^\ell) = 0$, and $\tilde{\mathcal{O}}(T^{-\frac{2\beta}{2\beta+1}})$ otherwise. However, for any finite T the rate of convergence is $\tilde{\mathcal{O}}(T^{-\frac{2\beta}{\beta+1}})$ for any $T = \mathcal{O}(\mathcal{E}^\ell(f_\rho^\ell)^{-\frac{\beta+1}{2\beta}})$. In other words, we can expect a first regime at faster convergence, that saturates when the number of samples becomes big enough. This is particularly important because often in practical applications the features and the kernel are chosen to have good performance that is low optimal ℓ-risk. Using standard excess risk comparison results, we can also obtain convergence results for the misclassification loss [2].

Regarding the optimality of our results, lower bounds for the square loss are known [17] under assumption (2) and further assuming that the eigenvalues of L_K have a polynomial decay, that is

$$(\lambda_i)_{i \in \mathbb{N}} \sim i^{-b}, \; b \geq 1. \tag{3}$$

Condition (3) can be interpreted as an effective dimension of the space. It always holds for $b = 1$ [17] and this is the condition we consider that is usually denoted as *capacity independent*, see the discussion in [19]. In the capacity independent setting, the lower bound is $\mathcal{O}(T^{-\frac{2\beta}{2\beta+1}})$, that matches the asymptotic rates in Theorem 1, up to logarithmic terms. Even if we require the loss function to be Lipschitz and smooth, it is unlikely that different lower bounds can be proved in our setting. Note that the lower bounds are worst case w.r.t. $\mathcal{E}^\ell(f_\rho^\ell)$, hence they do not cover the case $\mathcal{E}^\ell(f_\rho^\ell) = 0$, where we get even better rates.

4 Related Work

The approach of stochastically minimizing the ℓ-risk of the square loss in a RKHS has been pioneered by [15]. The rates were improved, but still suboptimal, in [20], with a general approach for locally Lipschitz loss functions in the origin. The optimal bounds, matching the ones we obtain for $\mathcal{E}^\ell(f_\rho^\ell) \neq 0$, were obtained for $\beta > 0$ in expectation by [19]. Their rates also hold for $\beta > \frac{1}{2}$, while our rates, as the ones in [17], saturate at $\beta = \frac{1}{2}$. In [18], high probability bounds were proved in the case that $\frac{1}{2} \leq \beta \leq 1$. Note that, while in the range $\beta \geq \frac{1}{2}$, that implies $f_\rho \in \mathcal{H}_K$, it is possible to prove high probability bounds [3, 17, 18], the range $0 < \beta < \frac{1}{2}$ considered in this paper is very tricky, see the discussion in [17]. In this range no high probability bounds are known without additional assumptions. All the previous approaches require the knowledge of β, while our algorithm is parameter-free. Also, we obtain faster rates for the excess ℓ-risk, when $\mathcal{E}^\ell(f_\rho^\ell) = 0$.

In the batch setting and square loss, the same optimal rates, but in high probability, were obtained by [3] for $\beta > \frac{1}{2}$, and by [17] in the range $0 < \beta \leq \frac{1}{2}$ using an additional assumption on the functions in $\mathcal{H}_K$. Again, these approaches require the knowledge of β or a cross-validation procedure.

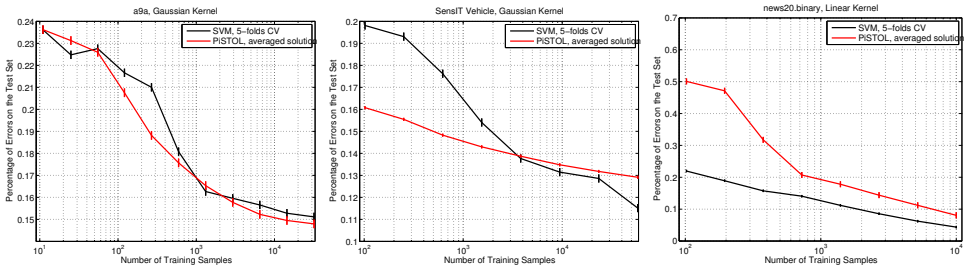

Figure 2: Average test errors and standard deviations of PiSTOL and SVM w.r.t. the number of training samples over 5 random permutations, on *a9a*, *SensIT Vehicle*, and *news20.binary*.

Table 1: The table shows the average logistic loss after one pass over the data (as called progressive validation error) using a linear kernel.

Dataset	# samples/features	VW	PiSTOL Per-coordinate
a9a	32,561/123	**0.3319**	0.3412
covtype.binary	581,012/54	0.5525	**0.5370**
ijcnn1	49,990/22	0.2009	**0.1983**
new20.binary	19,996/1,355,191	0.2868	**0.1734**
real-sim	72,309/20,958	0.1247	**0.1190**
url	2,396,130/3,231,961	0.04999	**0.03635**

5 Empirical Results

As a proof of concept on the potentiality of this method we have also run few preliminary experiments, to compare the performance of PiSTOL to a kernel SVM using 5-folds cross-validation to select the regularization weight parameter. The experiments were repeated with 5 random shuffles, showing the average and standard deviations over three datasets.[3] The latest version of LIBSVM was used to train the SVM [4]. We have that PiSTOL closely tracks the performance of the tuned SVM when a Gaussian kernel is used. Also, contrary to the common intuition, the stochastic approach of PiSTOL seems to have an advantage over the tuned SVM when the number of samples is small. Probably, cross-validation is a poor approximation of the generalization performance in that regime, while the small sample regime does not affect at all the analysis of PiSTOL. Note that in the case of News20, a linear kernel is used over the vectors of size 1355192. The finite dimensional case is not covered by our theorems, still we see that PiSTOL seems to converge at the same rate of SVM, just with a worse constant. It is important to note that the total time the 5-folds cross-validation plus the training with the selected parameter for the SVM on 58000 samples of *SensIT Vehicle* takes ~ 6.5 hours, while our unoptimized Matlab implementation of PiSTOL less than 1 hour, ~ 7 times faster. The gains in speed are similar on the other two datasets.

We also tested the performance of a per-coordinate variant of PiSTOL, suitable for linear kernels in finite dimensional spaces, implementing it in the Vowpal Wabbit (VW) software[4]. In this case, we do not have the guarantees of the standard version of PiSTOL, but we can still prove a worst case regret bound, see [10] for details. We compared it to the performance of VW, the de-facto standard to train linear classifiers in industry. As shown in [7], per-coordinate online algorithms for convex losses can be very easily designed and analyzed just running an independent copies of the algorithm on each coordinate, each one with their own parameters $\alpha_{i,t}, a_{i,t}, b_{i,t}$. Also, for each coordinate i we set $a_{i,t} = L \max_{j \le t} |x_{i,j}|$ and $b_{i,t} = 0.5\sqrt{\alpha_{i,t}}$. This choice gives a scale-free algorithm, in the sense that the predictor is independent of arbitrary scalings of the coordinates of the gradients of the loss function, similar to the algorithms in [11, 13]. We used the logistic loss and did not tune the parameters of VW, to evaluate the ability of the algorithms to self-tune theirs parameters. The results are presented in Table 1. The training times of our implementation are the same of VW.

[3] Datasets available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. The precise details to replicate the experiments are in [10].

[4] `https://github.com/JohnLangford/vowpal_wabbit/wiki`

Acknowledgments

I am thankful to Lorenzo Rosasco for introducing me to the beauty of the operator L_K^β and to Brendan McMahan for fruitful discussions.

References

[1] F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In *NIPS*, pages 773–781, 2013.

[2] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, Classification, and Risk Bounds. *Journal of the American Statistical Association*, 101(473):138–156, March 2006.

[3] A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[5] D.-R. Chen, Q. Wu, Y. Ying, and D.-X. Zhou. Support vector machine soft margin classifiers: Error analysis. *Journal of Machine Learning Research*, 5:1143–1175, 2004.

[6] F. Cucker and D. X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, New York, NY, USA, 2007.

[7] H. B. McMahan M. Streeter. Less regret via online conditioning, 2010. arXiv:1002.4862.

[8] H. B. McMahan and F. Orabona. Unconstrained online linear learning in Hilbert spaces: Minimax algorithms and normal approximations. In *COLT*, 2014.

[9] F. Orabona. Dimension-free exponentiated gradient. In *Advances in Neural Information Processing Systems 26*, pages 1806–1814. Curran Associates, Inc., 2013.

[10] F. Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning, 2014. arXiv:1406.3816.

[11] F. Orabona, K. Crammer, and N. Cesa-Bianchi. A generalized online mirror descent with applications to classification and regression. *Machine Learning Journal*, In print, 2014. arXiv:1304.2994.

[12] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, 22:400–407, 1951.

[13] S. Ross, P. Mineiro, and J. Langford. Normalized online learning. 2013.

[14] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proc. of ICML*, pages 807–814, 2007.

[15] S. Smale and Y. Yao. Online learning algorithms. *Found. Comp. Math*, 6:145–170, 2005.

[16] N. Srebro, K. Sridharan, and A. Tewari. Smoothness, low noise and fast rates. In *Advances in Neural Information Processing Systems 23*, pages 2199–2207. Curran Associates, Inc., 2010.

[17] I. Steinwart, D. R. Hush, and C. Scovel. Optimal rates for regularized least squares regression. In *COLT*, 2009.

[18] P. Tarrès and Y. Yao. Online learning as stochastic approximation of regularization paths, 2013. arXiv:1103.5538.

[19] Y. Ying and M. Pontil. Online gradient descent learning algorithms. *Foundations of Computational Mathematics*, 8(5):561–596, 2008.

[20] Y. Ying and D.-X. Zhou. Online regularized classification algorithms. *IEEE Trans. Inf. Theory*, 52(11):4775–4788, 2006.

[21] T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proc. of ICML*, pages 919–926, New York, NY, USA, 2004. ACM.