Parameter-Free Convex Learning through Coin Betting

Francesco Orabona FRANCESCO@ORABONA.COM

Dávid Pál DPAL@YAHOO-INC.COM

Yahoo Research, New York

Abstract

We present a new parameter-free algorithm for online linear optimization over any Hilbert space. It is theoretically optimal, with regret guarantees as good as with the best possible learning rate. The algorithm is simple and easy to implement. The analysis is given via the adversarial coin-betting game, Kelly betting and the Krichevsky-Trofimov estimator. Applications to obtain parameter-free convex optimization and machine learning algorithms are shown.

1. Introduction

We consider the Online Linear Optimization (OLO) (Cesa-Bianchi and Lugosi, 2006; Shalev-Shwartz, 2011) over a Hilbert space $\mathcal{H}$. In each round t, an algorithm chooses a point $w_t \in \mathcal{H}$ and then receives a loss vector $\ell_t \in \mathcal{H}$. The algorithm's goal is to keep its *regret* small, defined as the difference between its cumulative reward and the cumulative reward of a fixed strategy $u \in \mathcal{H}$, that is

$$\text{Regret}_T(u) = \sum_{t=1}^{T} \langle \ell_t, w_t \rangle - \sum_{t=1}^{T} \langle \ell_t, u \rangle \ .$$

where $\langle \cdot, \cdot \rangle$ is the inner product in $\mathcal{H}$.

OLO is a basic building block of many machine learning problems. For example, Online Convex Optimization (OCO) is a problem analogous to OLO where the linear function $u \mapsto \langle \ell_t, u \rangle$ is generalized to an arbitrary convex function $f_t(u)$. OCO is solved through a reduction to OLO by feeding the algorithm $\ell_t = \nabla f_t(w_t)$ (Shalev-Shwartz, 2011). Batch and stochastic convex optimization can also be solved through a reduction to OLO by taking the average of $w_1, w_2, \ldots, w_T$ (Shalev-Shwartz, 2011).

To achieve optimal regret, most of the existing online algorithms (e.g. Online Gradient Descent, Hedge) require the user to set the learning rate to an unknown/oracle value. Recently, new parameter-free algorithms have been proposed for OLO/OCO (Chaudhuri et al., 2009; Chernov and Vovk, 2010; Streeter and McMahan, 2012; Orabona, 2013; McMahan and Abernethy, 2013; McMahan and Orabona, 2014; Luo and Schapire, 2014; Orabona, 2014; Luo and Schapire, 2015; Koolen and van Erven, 2015). These algorithms adapt to the characteristics of the optimal predictor, without the need to tune parameters. However, their *design and underlying intuition* is still a challenge.

Our contributions are as follows. We connect algorithms for OLO with coin betting. Namely, we show that an algorithm for OLO can be viewed as an algorithm for betting on outcomes of adversarial coin flips. The wealth the algorithm can generate for the betting problem is connected to the regret in OLO setting. This insight allows us to design novel parameter-free algorithms,

which are extremely simple and natural. We also show some applications of our results to convex optimization and machine learning, as well as some empirical results.

2. How to Tune the Learning Rates?

Denote by $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$ the induced norm in $\mathcal{H}$ and assume that $\|\ell_t\| \leq 1$. Consider OLO over a Hilbert Space $\mathcal{H}$. Online Gradient Descent (OGD) with learning rate η satisfies (Shalev-Shwartz, 2011)

$$\forall u \in \mathcal{H} \qquad \text{Regret}_T(u) \leq \frac{\|u\|^2}{2\eta} + \frac{\eta T}{2} \,. \tag{1}$$

It is obvious that the optimal tuning of the learning rate depends on the unknown norm of u.

The simple choice $\eta = 1/\sqrt{T}$ leads to an algorithm that satisfies

$$\forall u \in H \qquad \text{Regret}_T(u) \leq \frac{1}{2}\left(1 + \|u\|^2\right)\sqrt{T} \,. \tag{2}$$

However, in this bound the dependency on $\|u\|$ is suboptimal: The quadratic dependency can be replaced by an (almost) linear dependency. Starting from (1), if we choose the learning rate $\eta = D/\sqrt{T}$, we get a family of algorithms parameterized by $D \in [0, \infty)$ that satisfy

$$\forall u \in \mathcal{H} : \|u\| \leq D \quad \Longrightarrow \quad \text{Regret}_T(u) \leq D\sqrt{T} \,. \tag{3}$$

Instead of a family of algorithms parameterized by $D \in [0, \infty)$ satisfying the bound (3), one *would like to have* a single algorithm (without any tuning parameters) satisfying

$$\forall u \in \mathcal{H} \qquad \text{Regret}_T(u) \leq \|u\|\sqrt{T} \,. \tag{4}$$

Notice that (4) is stronger than (3) in the following sense: A single algorithm satisfying (4) implies (3) for all values of $D \in [0, \infty)$. However, a family of algorithms $\{A_D : D \in [0, \infty)\}$ parameterized by D where A_D satisfies (3), does not yield a single algorithm that satisfies (4). Finally, note that (4) has better dependency on $\|u\|$ than (2).

Better guarantees are indeed possible. In fact, there have been a lot of work on algorithms (Streeter and McMahan, 2012; Orabona, 2013; McMahan and Abernethy, 2013; McMahan and Orabona, 2014; Orabona, 2014) that satisfy a slightly weaker version of (4). Namely, their regret satisfies

$$\forall u \in \mathcal{H} \qquad \text{Regret}_T(u) \leq \left(O(1) + \text{polylog}(1 + \|u\|)\,\|u\|\,\right)\sqrt{T} \,. \tag{5}$$

It can be shown that for OLO over Hilbert space the extra poly-logarithmic factor is necessary (McMahan and Abernethy, 2013; Orabona, 2013). Algorithms satisfying (5) are called *parameter-free*, since they do not need to know D, yet they have an optimal dependency on $\|u\|$.

3. Parameter-Free Algorithm From Coin Betting

Here, we present our new parameter-free algorithm for OLO over a Hilbert space $\mathcal{H}$, stated as Algorithm 1. We would like to stress the extreme simplicity of the algorithm. The theorem below upper bounds its regret in the form of (5), the proof can be found in Orabona and Pál (2016).

Algorithm 1 Algorithm for OLO over Hilbert space $\mathcal{H}$ based on Krichevsky-Trofimov estimator

for $t = 1, 2, \dots$ **do**
  Predict with $w_t \leftarrow -\frac{1}{t}\left(1 - \sum_{i=1}^{t-1}\langle \ell_i, w_i\rangle\right)\sum_{i=1}^{t-1}\ell_i$
  Receive loss vector $\ell_t \in \mathcal{H}$ such that $\|\ell_t\| \le 1$
end for

Theorem 1 (Regret Bound for Algorithm 1) *Let $\{\ell_t\}_{t=1}^{\infty}$ be any sequence of loss vectors in a Hilbert space $\mathcal{H}$ such that $\|\ell_t\| \le 1$. Algorithm 1 satisfies*

$$\forall T \ge 0 \quad \forall u \in \mathcal{H} \qquad \text{Regret}_T(u) \le \|u\| \sqrt{T \ln\left(1 + 4T^2 \|u\|^2\right)} + 1 \,.$$

We now explain how Algorithm 1 is derived from the Krichevsky-Trofimov solution to the adversarial coin-betting problem.

Adversarial Coin Betting. Consider a gambler making repeated bets on the outcomes of adversarial coin flips. The gambler starts with an initial endowment of 1 dollar. In each round t, he bets on the outcome of a coin flip $c_t \in \{-1, 1\}$, where $+1$ denotes heads and -1 denotes tails. The outcome c_t is chosen by an adversary. The gambler can bet any amount on either heads or tails. However, he cannot borrow any additional money. If he loses, he loses the betted amount; if he wins, he gets the betted amount back and, in addition to that, he gets the same amount as a reward. We encode the gambler's bet in round t by a single number $\beta_t \in [-1, 1]$. The sign of β_t encodes whether he is betting on heads or tails. The absolute value encodes the betted amount as the fraction of his current wealth. Let Wealth_t be gambler's wealth at the end of round t. It satisfies

$$\text{Wealth}_0 = 1 \qquad \text{and} \qquad \text{Wealth}_t = (1 + c_t\beta_t)\,\text{Wealth}_{t-1} \qquad \text{for } t \ge 1 \,. \tag{6}$$

Note that since $\beta_t \in [-1, 1]$, gambler's wealth stays always non-negative.

Kelly Betting and Krichevsky-Trofimov Estimator. For sequential betting on i.i.d. coin flips, the optimal strategy has been proposed by Kelly (1956). The strategy assumes that the coin flips $\{c_t\}_{t=1}^{\infty}$, $c_t \in \{+1, -1\}$, are generated i.i.d. with known probability of heads. If $p \in [0, 1]$ is the probability of heads, the Kelly bet is $\beta_t = 2p - 1$. He showed that, in the long run, this strategy will provide more wealth than betting any other fixed fraction (Kelly, 1956).

For adversarial coins, Kelly betting does not make sense. Krichevsky and Trofimov (1981) proposed to replace p with an estimate: After seeing coin flips $c_1, c_2, \dots, c_{t-1}$, use the empirical estimate $k_t = \frac{1/2 + \sum_{i=1}^{t-1}\mathbf{1}[c_i = +1]}{t}$. Their estimate is commonly called *KT estimator*[1] and it results in the betting strategy $\beta_t = 2k_t - 1 = \frac{\sum_{i=1}^{t-1} c_i}{t}$. Krichevsky and Trofimov showed that this strategy guarantees almost the same wealth that one would obtain knowing in advance the fraction of heads. Namely, if we denote by $\text{Wealth}_t(\beta)$ the wealth of the strategy that bets the fraction β in every round, then the wealth of the Krichevsky-Trofimov betting strategy satisfies

$$\forall \beta \in [-1, 1] \qquad \text{Wealth}_t \ge \frac{\text{Wealth}_t(\beta)}{2\sqrt{t}} \,. \tag{7}$$

Moreover, this guarantee is optimal up to constant multiplicative factors (Cesa-Bianchi and Lugosi, 2006).

1. Compared to the standard maximum likelihood estimate $\frac{\sum_{i=1}^{t-1}\mathbf{1}[c_i = +1]}{t-1}$, KT estimator "shrinks" slightly towards $\frac{1}{2}$.

Algorithm 2 SGD algorithm based on KT estimator

Require: Convex functions $f_1, f_2, \dots, f_N$ and desired number of iterations T

Initialize $\text{Wealth}_0 \leftarrow 1$ and $\theta_0 \leftarrow 0$
for $t = 1, 2, \dots, T$ **do**
  Set $w_t \leftarrow \text{Wealth}_{t-1} \frac{\theta_{t-1}}{t}$
  Select an index j at random from $\{1, 2, \dots, N\}$ and compute $\ell_t = \nabla f_j(w_{t-1})$
  Update $\theta_t \leftarrow \theta_{t-1} - \ell_t$ and $\text{Wealth}_t \leftarrow \text{Wealth}_{t-1} - \langle \ell_t, w_t \rangle$
end for
Output $\overline{w}_T = \frac{1}{T} \sum_{t=1}^{T} w_t$

From betting to OLO. In Algorithm 1, the "coin outcome" is the vector $c_t \in \mathcal{H}$ where $c_t = -\ell_t$ and algorithm's wealth is $\text{Wealth}_t = 1 + \sum_{i=1}^{t} \langle c_i, w_i \rangle = 1 - \sum_{i=1}^{t} \langle \ell_i, w_i \rangle$. The algorithm explicitly keeps track of its wealth and it bets "vectorial fraction" $\beta_t = \frac{\sum_{i=1}^{t-1} c_i}{t} = -\frac{\sum_{i=1}^{t-1} \ell_i}{t}$ of its current wealth. The regret bound (Theorem 1) is a consequence of Krichevsky-Trofimov lower bound (7) on the wealth and the duality between regret and wealth. For more details, see Orabona and Pál (2016).

4. From Online Learning to Convex Optimization and Machine Learning

The result in Section 3 immediately implies new algorithms and results in convex optimization and machine learning. We will state some of them here, see Orabona (2014) for more results.

Convex Optimization. Consider an empirical risk minimization problem of the form

$$F(w) = \frac{1}{N} \sum_{i=1}^{N} f_i(w), \tag{8}$$

where $f_i : \mathbb{R}^d \to \mathbb{R}$ is convex.[2] It is immediate to transform Algorithm 1 into a Stochastic Gradient Descent (SGD) algorithm for this problem, obtaining Algorithm 2. In Algorithm 2, $\nabla f_j(w)$ denotes a subgradient of f_j at a point w. We assume that the norm of the subgradient of f_j is bounded by 1.

Beside the simplicity of the Algorithm 2, it has the important property is that it *does not have a learning rate to be tuned*, yet it achieves the optimal convergence rate. In fact, denoting by $\widehat{w} = \arg\min_w F(w)$ the optimal solution of (8), the following theorem states the rate of convergence of Algorithm 2.

Theorem 2 *The average $\overline{w}_T$ produced by Algorithm 2 is an approximate minimizer of the objective function (8):*

$$\mathbf{E}\left[F(\overline{w}_T)\right] - F(\widehat{w}) \leq \frac{\|\widehat{w}\|}{\sqrt{T}} \sqrt{\log(1 + 4T^2 \|\widehat{w}\|^2)} + \frac{1}{T} .$$

Note that in the above theorem, T can be larger (multiple epochs) or smaller than N.

Machine Learning. In machine learning, the minimization of a function (8) is just a proxy to minimize the *true risk* over an unknown distribution. For example, $f_i(w)$ can be of the form $f_i(w) = f(w, X_i, Y_i)$ where $\{(X_i, Y_i)\}_{i=1}^{N}$ is a sequences of labeled sampled generated i.i.d. from some *unknown* distribution and $f(w, X_i, Y_i)$ is the logistic loss of a weight vector w on a sample

2. The algorithm can also be implemented and analyzed with kernels (Orabona, 2014).

Algorithm 3 Averaging algorithm based on KT estimator

Require: Sample $(X_1, Y_1), (X_2, Y_2), \dots, (X_N, Y_N)$
Initialize $\text{Wealth}_0 \leftarrow 1$ and $\theta_0 \leftarrow 0$
for $i = 1, 2, \dots, N$ **do**
  Set $w_i \leftarrow \text{Wealth}_{i-1} \frac{\theta_{i-1}}{i}$
  Compute $\ell_i = \frac{\partial f(w, X_i, Y_i)}{\partial w}|_{w=w_i}$
  Update $\theta_i \leftarrow \theta_{i-1} - \ell_i$ and $\text{Wealth}_i \leftarrow \text{Wealth}_{i-1} - \langle \ell_i, w_i \rangle$
end for
Output $\overline{w}_N = \frac{1}{N} \sum_{i=1}^{N} w_i$

(X_i, Y_i). A common approach to have a small risk on the test set is to minimize a regularized objective function over the training set:

$$F_\lambda^{\text{Reg}}(w) = \lambda \|w\|^2 + \frac{1}{N} \sum_{i=1}^{N} f(w, X_i, Y_i) \ . \tag{9}$$

This problem is strongly convex, so there are very efficient methods to minimize it, hence we can assume to be able to get the minimizer of F_λ^{Reg} with arbitrary high precision and algorithms that do not require to tune learning rates. Yet, this is not enough. In fact, we are rarely interested in the value of the objective function F_λ^{Reg} or its minimizer, rather we are interested in the *true risk* of a solution w, that is $\mathbf{E}[f(w, X, Y)]$, where (X, Y) is an independent "test" sample from the same distribution from which the training set $\{(X_i, Y_i)\}_{i=1}^N$ came from. Hence, in order to get a good performance we have to select a good regularization parameter. In particular, from Sridharan et al. (2009) we get

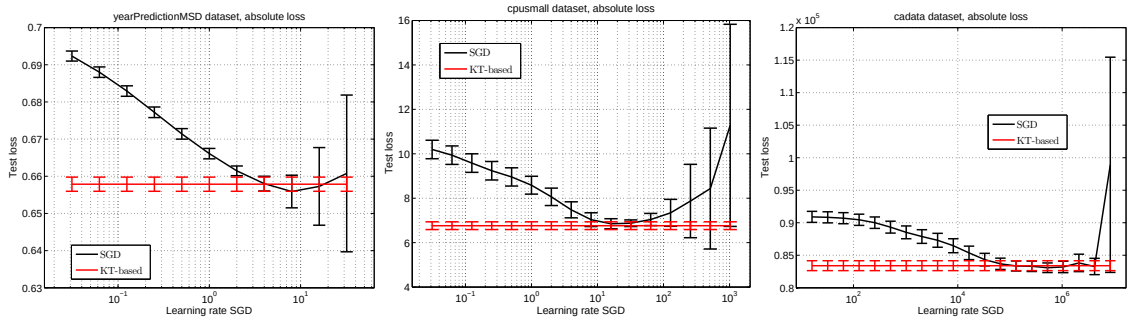
$$\mathbf{E}[f(\widehat{w}_\lambda, X, Y)] - \mathbf{E}[f(w^*, X, Y)] \leq O(\lambda \|w^*\|^2 + \tfrac{1}{\lambda N}) \ ,$$

where $w^* = \arg\min_w \mathbf{E}[f(w, X, Y)]$ and $\widehat{w}_\lambda = \arg\min_w F_\lambda^{\text{Reg}}(w)$. From the above bound, it is clear that the optimal value of λ depends on the $\|w^*\|$ that is unknown. Yet another possibility is to select the optimal learning rate and/or the number of epochs of SGD to directly minimize $\mathbf{E}[f(w^*, X, Y)]$. However, all these methods are equivalent (J. Lin, 2016) and they still require to tune at least one parameter. We would like to stress that this is not just a theoretical problem: Any practitioner knows how painful it is to find the right regularization for the problem at hand.

Assuming we would know $\|w^*\|$, we could set $\lambda = O(1/(\|w^*\| \sqrt{N}))$ to achieve the worst-case optimal bound

$$\mathbf{E}[f(\widehat{w}_\lambda, X, Y)] - \mathbf{E}[f(w^*, X, Y)] \leq O\left(\frac{\|w^*\|}{\sqrt{N}}\right) \ . \tag{10}$$

However, we can get the same guarantee without knowing $\|w^*\|$ or the optimal λ, by doing a single pass over the data set. More precisely, we derive Algorithm 3 from Algorithm 1 by applying the standard online-to-batch reduction (Shalev-Shwartz, 2011). The algorithm makes only a single pass over the dataset and it does not have any tuning parameters. Yet, it has almost the same guarantee (10) *without knowing* $\|w^*\|$ *or the optimal regularization parameter* λ *or the learning rate, or any other tuning parameter*.



(*a*) (*b*) (*c*)

Figure 1: Test loss versus learning rate parameter of SGD (in log scale), compared with the parameter-free Algorithm 3.

Theorem 3 *Assume that* $(X,Y),(X_1,Y_1),(X_2,Y_2),\dots,(X_N,Y_N)$ *are i.i.d. The output* $\overline{w}_N$ *of Algorithm 3 satisfies*

$$\mathbf{E}[f(\overline{w}_N,X,Y)]-\mathbf{E}[f(w^*,X,Y)]\leq\frac{\|w^*\|}{\sqrt{N}}\sqrt{\log(1+4N^2\|w^*\|^2)}+\frac{1}{N}\ .$$

Comparing this guarantee to the one in (10), we see that, just paying a sub-logarithmic price, we obtain the optimal convergence rate and we remove all the parameters.

5. Empirical Evaluation

We have also run a small empirical evaluation to show that the theoretical difference between classic learning algorithms and parameter-free ones is real and not just theoretical. In Figure 1, we have used three regression datasets,[3] and solved the OCO problem through OLO. In all the three cases, we have used the absolute loss and normalized the input vectors to have L2 norm equal to 1.

The dataset were split in two parts: 75% training set and the remaining as test set. The training is done through one pass over the training set and the final classifier is evaluated on the test set. We used 5 different splits of training/test and we report average and standard deviations.

We have run SGD with different learning rates and shown the performance of its last solution on the test set. For Algorithm 3, we do not have any parameter to tune so we just plot its test set performance as a line.

From the empirical results, it is clear that the optimal learning rate is completely data-dependent. It is also interesting to note how the performance of SGD becomes very unstable with large learning rates. Yet *our parameter-free algorithm has a performance very close to the unknown optimal tuning of the learning rate of SGD*.

Acknowledgments

The authors thank Jacob Abernethy, Nicolò Cesa-Bianchi, Satyen Kale, Chansoo Lee, and Giuseppe Molteni for useful discussions on this work.

3. Datasets available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

References

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

K. Chaudhuri, Y. Freund, and D. Hsu. A parameter-free hedging algorithm. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22 (NIPS 2009), December 7-12, Vancouver, BC, Canada*, pages 297–305, 2009.

A. Chernov and V. Vovk. Prediction with advice of unknown number of experts. In Peter Grünwald and Peter Spirtes, editors, *Proc. of the 26th Conf. on Uncertainty in Artificial Intelligence (UAI 2010), July 8-11, Catalina Island, California, USA*. AUAI Press, 2010.

L. Rosasco J. Lin, R. Camoriano. Generalization properties and implicit regularization for multiple passes SGM. In *Proc. of International Conference on Machine Learning (ICML)*, 2016.

J. L. Kelly. A new interpretation of information rate. *Information Theory, IRE Transactions on*, 2(3):185–189, September 1956.

W. M. Koolen and T. van Erven. Second-order quantile methods for experts and combinatorial games. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proc. of the 28th Conf. on Learning Theory (COLT 2015), July 3-6, Paris, France*, pages 1155–1175, 2015.

R. E. Krichevsky and V. K. Trofimov. The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2):199–206, 1981.

H. Luo and R. E. Schapire. A drifting-games analysis for online learning and applications to boosting. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1368–1376. Curran Associates, Inc., 2014.

H. Luo and R. E. Schapire. Achieving all with no parameters: AdaNormalHedge. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proc. of the 28th Conf. on Learning Theory (COLT 2015), July 3-6, Paris, France*, pages 1286–1304, 2015.

H. B. McMahan and J. Abernethy. Minimax optimal algorithms for unconstrained linear optimization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 2724–2732, 2013.

H. B. McMahan and F. Orabona. Unconstrained online linear learning in Hilbert spaces: Minimax algorithms and normal approximations. In Vitaly Feldman, Csaba Szepesvri, and Maria Florina Balcan, editors, *Proc. of The 27th Conf. on Learning Theory (COLT 2014), June 13-15, Barcelona, Spain*, pages 1020–1039, 2014.

F. Orabona. Dimension-free exponentiated gradient. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1806–1814. Curran Associates, Inc., 2013.

F. Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems 27*, pages 1116–1124, 2014.

F. Orabona and D. Pál. From coin betting to parameter-free online learning, 2016. Available from: http://arxiv.org/pdf/1602.04128.pdf.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

K. Sridharan, S. Shalev-Shwartz, and N. Srebro. Fast rates for regularized objectives. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1545–1552. Curran Associates, Inc., 2009.

M. Streeter and B. McMahan. No-regret algorithms for unconstrained online convex optimization. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25 (NIPS 2012), Decmber 3-8, Lake Tahoe, Nevada, USA*, pages 2402–2410, 2012.