# A Summary of Fast Convergence of Regularized Learning in Games [7]

Malte Schledjewski

18th August 2016

## 1 Games and Learning

Games can model many interesting scenarios like auctions or routing packets in networks. If you do not know the best way to play the game or you do not know how the others will behave you have to learn how to play it on the fly. This is the setting for online learning in games.

### 1.1 A Generous Game Model

The paper uses the following game model: The game is played in rounds. There is a finite amount of players. Each player $i$ has a set $S_i$ of possible strategies that he can play. All strategies sets have the same finite cardinality. Each player has a normalized utility function $u_i : S_1 \times \ldots \times S_n \to [0, 1]$. $\mathbf{w}_i$ denotes a mixed strategy of player $i$ which is a vector containing for each strategy the probability that this strategy is played. All entries therefore sum up to 1.

In each round $t$ each player $i$ chooses a mixed strategy $\mathbf{w}_i^t$. At the end of the round each player observes the vector $\mathbf{u}_i^t = (u_{i,x}^t)_{x \in S_i}$ which states for each strategy he could have played the expected utility he would have gained in this round. The (expected) utility gained by the player in this round is therefore simply the inner product $\langle \mathbf{w}_i^t, \mathbf{u}_i^t \rangle$. The observation of $\mathbf{u}_i^t$ makes the game quite generous in a sense because it gives a lot of information. In a real world scenario, like an auction, you would probably only get to know the played strategies of all the players.

After $T$ time steps each player has regret

$$r_i(T) = \sup_{\mathbf{w}_i^* \in \Delta(S_i)} \sum_{t=1}^{T} \left\langle \mathbf{w}_i^* - \mathbf{w}_i^t, \mathbf{u}_i^t \right\rangle .$$

which is defined as the maximum gain he could have achieved by switching to any other fixed mixed strategy $\mathbf{w}_i^*$. An algorithm for choosing the mixed strategy is called a no-regret algorithm if it has vanishing regret which means $r_i(T) = o(T)$ and therefore the regret averaged over all rounds converges to 0. In the paper it is assumed that all players use a no-regret algorithm. This constraint is not that tight because no-regret algorithms are a natural choice because they have a regret convergence rate of $O(1/\sqrt{T})$ even against adversarial environments where this bound is unimprovable.

This game model maps naturally to the Expert Advise Framework which is commonly used in online learning. Shalev-Shwartz [6] gives an overview of online learning. He also explains why probabilities are chosen instead of choosing a single strategy. It is the more general technique to avoid the problem that otherwise an adversary can always make the prediction wrong and the utility very low. He also covers that by using the probabilities the problem is transformed into an online convex optimization problem. This connection will appear in the two studied algorithms.

## 1.2 Two No-Regret Algorithms

The paper takes a closer look at two well-known algorithms. Even the optimistic versions exist since 2012 [2].

### 1.2.1 Optimistic Follow the Regularized Leader

The basic intuition for this algorithm is based on the definition of regret. One looks at the best fixed mixed strategy up until now to determine the regret. So simply uses this mixed strategy, which was optimal until now, for the next round. This is known as Follow the Leader.

This idea has a problem because it is unstable. There is a simple example sequence which alternates and Follow the Leader lags behind one step and therefore predicts exactly the opposite [6, p.127, Example 2.2]. So called regularizers are introduced to make the system stable. A penalty term based on a regularizer is added to the maximisation term to punish 'complex' mixed strategies [5, p.89]. This adapted version is then called Follow the Regularized Leader. There are many different regularizers, one being the entropic regularizer [6, p.136, Example 2.5]. Regularizers are also known in typical machine learning scenarios to prevent the system from learning the noise in the training data. They are often used to enforce sparsity in the result.

Optimistic Follow the Regularized Leader (OFTRL) [2] simply tries to predict the expected utility vector $\mathbf{u}_i^{t+1}$ for the next round by using an adaptive prediction sequence. The maximisation term now also contains the predicted next round. OFTRL is defined by the choice of the regularizer, the adaptive prediction sequence and the so called stepsize $\eta$ which determines how much weight the regularization has.

### 1.2.2 Optimistic Mirror Descent

Mirror descent is closely connected to Follow the Regularized Leader. It is also closely connected to online convex optimization. This connection also explains why the regularizer for both algorithms has to be 1-strongly convex (see [6] for a detailed explanation). Mirror descent basically is projected gradient descent. Simple gradient descent is not applicable because it may leave the probability simplex.

The Optimistic Mirror Descent (OMD) [2] also adds an adaptive prediction sequence. OMD is easier to compute because the iteration step does not entail solving a maximization problem.

# 2 Favourable Environments

The best possible regret convergence rate in adversarial environments is $O(1/\sqrt{T})$. But how much can this be improved if the environment is not adversarial but more predictable? For two-player zero-sum games it is already known that Optimistic Mirror Decent achieves a rate of $O(1/T)$ under suitable conditions if the last observed utility vector is used as the prediction for the next round [3]. There are also specificly adapted versions of algorithms that achieve the lower bound in favourable environments and fall back to the optimal rate of $O(1/\sqrt{T})$ in adversarial environments. But there is no general approach.

This paper generalizes these results by showing a broader class of algorithms for the mentioned game model and giving a black box reduction that achieves the good rate in adversarial environments.

## 2.1 `RVU` property

The paper introduces a new property which defines a class of algorithms:

**Definition 3** (RVU property)**.** We say that a vanishing regret algorithm satisfies the Regret bounded by Variation in Utilities (RVU) property with parameters $\alpha > 0$ and $0 < \beta \leq \gamma$ and a pair of dual norms $(\|\cdot\|, \|\cdot\|_*)$ if its regret on any sequence of utilities $\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^T$ is bounded as

$$\sum_{t=1}^{T} \left\langle \mathbf{w}^* - \mathbf{w}^t, \mathbf{u}^t \right\rangle \leq \alpha + \beta \sum_{t=1}^{T} \|\mathbf{u}^t - \mathbf{u}^{t-1}\|_*^2 - \gamma \sum_{t=1}^{T} \|\mathbf{w}^t - \mathbf{w}^{t-1}\|^2.$$

The left side of the inequality is simply the definition of regret. The dual to a norm $\|\cdot\|$ is defined as $\|\mathbf{u}\|_* = \sup_{\|\mathbf{w}\| \leq 1} \langle \mathbf{w}, \mathbf{u} \rangle$. The paper uses the RVU property only with $\|\cdot\|$ being any norm equivalent to the $\ell_1$ norm. The dual to the $\ell_1$ norm is the max norm. For a player $i$ the inequality becomes:

$$r(T) \leq \alpha + \beta \sum_{t=1}^{T} \max_{x \in S_i} \left| u_{i,x}^t - u_{i,x}^{t-1} \right|_1^2 - \gamma \sum_{t=1}^{T} \left\| \mathbf{w}_i^t - \mathbf{w}_i^{t-1} \right\|_1^2$$

At this point it should be stressed that $\alpha$, $\beta$ and $\gamma$ are terms and in general not constants. $\alpha = O(1)$ is assumed for the rest of the paper. The middle term accumulates the maximal change in expected utility for all possible strategies. This term is therefore small if the environment is stable so that utilities do not change a lot. In case of rapidly changing expected utilities it is possible to compensate this large term by the third term, which expresses the change in the played mixed strategy. The intuition is that if the utilities change a lot, it makes sense to alter the mixed strategy a lot. But in general the change of the played mixed strategy should rather be small because on the one hand it is possible to bound the regret in terms of change of the played mixed strategy [1, p.151, Lemma 7] and on the other hand if all players use similar algorithms, then the change of expected utilities is influenced by all other players' change of their played mixed strategy.

### 2.2 The Algorithms with Recency Bias

The paper shows that using recency bias for the prediction in both algorithms leads to fulfilment of the RVU property. Two adaptive prediction sequences are analysed: $H$-step recency bias, where the average of the last $H$ observed utility vectors is used as the prediction, and geometrically discounted recency bias.

The intuition is that over time the player learn to play the game, so it makes sense to use recent behaviour as a guideline and to ignore the behaviour that was shown in the beginning where no player knew how to play the game.

## 3 Social Welfare and Individual Regret

So now that we have this new class of algorithms what happens if all players use an algorithm from this class?

### 3.1 Social Welfare

Social welfare is the accumulated utility summed up over all players. The paper only explores social welfare in smooth games as defined by Roughgarden in [4]. Examples for smooth games are second-price auctions and congestion games. Roughgarden basically showed that in smooth games the social welfare converges to an approximate optimum. This convergences is driven by

the sum of all players' regret. It is therefore sufficient to show that this sum is constant bounded to achieve a convergence rate of $O(1/T)$ due to averaging over all time steps.

The paper uses the following theorem which does not depend on all players using the same algorithm but only on all used algorithms having the same parameters $\alpha$, $\beta$ and $\gamma$:

**Theorem 4.** Suppose that the algorithm of each player $i$ satisfies the property RVU with parameters $\alpha, \beta$ and $\gamma$ such that $\beta \leq \gamma/(n-1)^2$ and $\|\cdot\| = \|\cdot\|_1$. Then $\sum_{i \in N} r_i(T) \leq \alpha n$.

The proof is quite straight forward. Summing up the RVU property over all players one uses the additional restriction on $\beta$ to upper bound the summation of the middle and last term by 0. This is achieved mostly by exploiting that the utilities are normalised, Jensen's inequality for convex functions (in this case squaring) and some facts about total variation.

### 3.2 Individual Regret

It is nice to know that the social welfare reaches an approximate optimum in smooth games when all players use algorithms that satisfy the RVU property with suitable parameters but what about the individual results? Social welfare only says something about the average regret. All players use no-regret algorithms so their regret growth sublinear. Still the regret convergence rate could vary dramatically.

By also requiring all algorithm to be stable, one can show that the individual regret is in $O(T^{1/4})$ and the convergence rate therefore $O(T^{-3/4})$. The paper shows the following theorem:

**Theorem 11.** Suppose that the players use algorithms satisfying the RVU property with parameters $\alpha > 0, \beta > 0, \gamma \geq 0$. If we further have the stability property $\|\mathbf{w}_i^t - \mathbf{w}_i^{t+1}\| \leq \kappa$, then for any player $\sum_{t=1}^{T} \langle \mathbf{w}_i^* - \mathbf{w}_i^t, \mathbf{u}_i^t \rangle \leq \alpha + \beta\kappa^2(n-1)^2 T$.

The proof is similar to the one for social welfare. Together with their results concerning the RVU property parameters for OFTRL with one-step recency they get:

**Corollary 12.** If all players use the OFTRL algorithm with $\mathbf{M}_i^t = \mathbf{u}_i^{t-1}$ and $\eta = (n-1)^{-1/2} T^{-1/4}$, then we have $\sum_{t=1}^{T} \langle \mathbf{w}_i^* - \mathbf{w}_i^t, \mathbf{u}_i^t \rangle \leq (R+4)\sqrt{n-1} \cdot T^{1/4}$.

This could be generalised to the case where the other players use arbitrary no-regret algorithms that satisfy the RVU property and the stability condition with the same parameters.

## 4 Adversarial Environments

So far the paper has shown that for this scenario of suitable algorithms from the RVU class the convergence rates of all players' regret and the individual regret can be improved. But what happens if the environment becomes adversarial? Do the algorithms fall back to the rate of $O(1/T)$, which is optimal in this case?

A parametric version of the RVU poperty is introduced to achieve guarantees under any conditions:

**Definition 13** (RVU($\rho$) property)**.** We say that a parametric algorithm $\mathcal{A}(\rho)$ satisfies the Regret bounded by Variation in Utilities($\rho$) (RVU($\rho$)) property with parameters $\alpha, \beta, \gamma > 0$ and a pair of dual norms $(\|\cdot\|, \|\cdot\|_*)$ if its regret on any sequence of utilities $\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^T$ is bounded as

$$\sum_{t=1}^{T} \left\langle \mathbf{w}^* - \mathbf{w}^t, \mathbf{u}^t \right\rangle \leq \frac{\alpha}{\rho} + \rho\beta \sum_{t=1}^{T} \|\mathbf{u}^t - \mathbf{u}^{t-1}\|_*^2 - \frac{\gamma}{\rho} \sum_{t=1}^{T} \|\mathbf{w}^t - \mathbf{w}^{t-1}\|^2.$$

For OMD and OFTRL $\rho = \eta$.

Their black-box reduction uses such a tunable algorithm and employs the doubling trick based on the accumulated change in expected utilities (compare [6, p.130] for the doubling trick based on time-based epochs).

The black-box reduction works in epochs and uses such a tunable algorithm as a base algorithm. During one epoch, the same parameter is used as input for the base algorithm to play the rounds. An epoch ends if the accumulated change in expected utilities (the sum in the middle term of the `RVU` property) reaches a certain threshold. This threshold starts at one and is doubled each time an epoch ends. The used parameter is adjusted for each new epoch.

Due to the threshold being doubled for each epoch there are at most $log(T)$ many epochs and for each epoch there is the regret bound based on the `RVU`($\rho$) property. This black-box reduction preserves the `RVU` property and also guarantees a regret in $\tilde{O}(\sqrt{T})$ against any environment for a suitable initial parameter. The results for individual regret are also preserved with the stability condition becoming $\|\mathbf{w}_i^t - \mathbf{w}_i^{t-1}\| \leq \kappa\rho$.

# 5 Experimental Evaluation

Follow the Regularized Leader with the entropic regularizer (historically known as the Hedge algorithm) and its optimistic version with recency bias one were compared in an auction scenario which is known to be a smooth game. As seen in Figure 1 the optimstistic version has almost no regret whereas the classical version has a growth in $O(\sqrt{T})$. The convergence is therefore much faster in the optimistic version.

The stability which was also a key point concerning the `RVU` property and the convergence of individual regret can also be seen in Figure 2 for the optimistic version.

# 6 Conclusion

This paper introduced the `RVU` property and showed that it is sufficient for showing that all players' regret converges at the rate of $O(1/T)$ if all used algorithms hold this property for some suitable common parameters. In smooth games this even implies convergences towards an approximate optimum of social welfare. This property is also enough to show each player regret converges at the rate of $T^{-3/4}$ if all used algorithms have again suitable common parameters and fulfil the stability condition. Both these rates are improvements over $O(1/\sqrt{T})$.

The black-box reduction can preserve these rates under good conditions for all tunable algorithms that satisfy the `RVU`($\rho$) property while also giving good guarantees in adversarial environments.

These results are also based on a more general game model than the two-player zero-sum games, making it much more applicable.

Recency bias was identified to be sufficient to make the two well-known algorithms OMD and OFTRL satisfy these properties.

The `RVU` property is apparently only sufficient but not needed to achieve faster rates. The most interesting question is probably whether the rates can still be improved over $O(1/\sqrt{T})$ if the players do not gain so much information. What happens when only the played strategies of all other players can be observed? This would match the information available in real scenarios more closely.

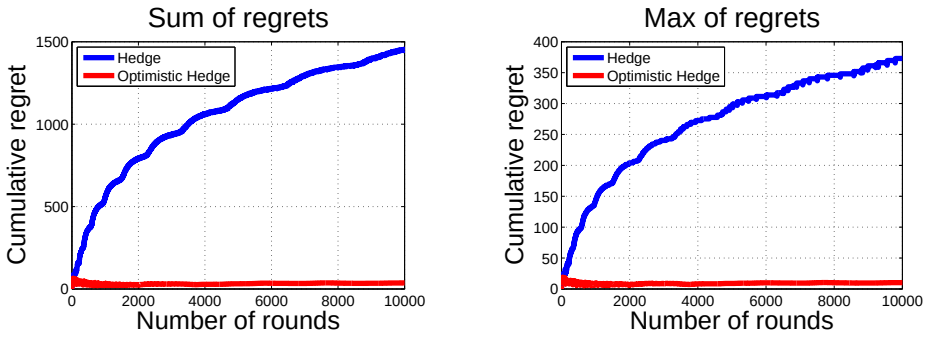


Figure 1: Maximum and sum of individual regrets over time under the Hedge (blue) and Optimistic Hedge (red) dynamics.

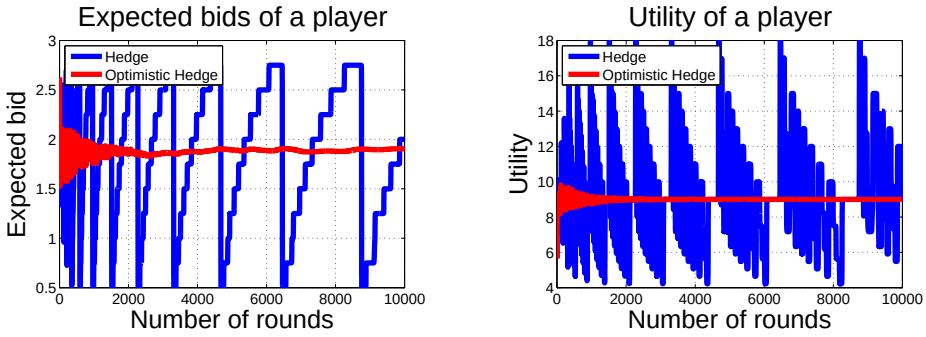


Figure 2: Expected bid and per-iteration utility of a player on one of the four items over time, under Hedge (blue) and Optimistic Hedge (red) dynamics.

# References


[1] Percy Liang. *CS229T/STAT231: Statistical Learning Theory (Winter 2016)*. Wed Apr 20 2016 01:36. Lecture Notes. Stanford University. 2016. URL: web.stanford.edu/class/cs229t/notes.pdf (visited on 17/08/2016).

[2] A. Rakhlin and K. Sridharan. 'Online Learning with Predictable Sequences'. In: *ArXiv e-prints* (Aug. 2012). arXiv: 1208.3728 [stat.ML].

[3] A. Rakhlin and K. Sridharan. 'Optimization, Learning, and Games with Predictable Sequences'. In: *ArXiv e-prints* (Nov. 2013). arXiv: 1311.1869 [cs.LG].

[4] Tim Roughgarden. 'Intrinsic Robustness of the Price of Anarchy'. In: *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*. STOC '09. Bethesda, MD, USA: ACM, 2009, pp. 513–522. ISBN: 978-1-60558-506-2. DOI: 10.1145/1536414.1536485. URL: http://doi.acm.org/10.1145/1536414.1536485.

[5] B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive computation and machine learning. MIT Press, 2002. ISBN: 9780262194754. URL: https://books.google.de/books?id=y8ORL3DWt4sC.

[6] Shai Shalev-Shwartz. 'Online Learning and Online Convex Optimization'. In: *Foundations and Trends® in Machine Learning* 4.2 (2012), pp. 107–194. ISSN: 1935-8237. DOI: 10.1561/2200000018. URL: http://dx.doi.org/10.1561/2200000018.

[7] V. Syrgkanis et al. 'Fast Convergence of Regularized Learning in Games'. In: *ArXiv e-prints* (July 2015). arXiv: 1507.00407 [cs.GT].