

Online Prediction, Low Regret, and Convex Duality

Shai Shalev-Shwartz

Toyota Technological Institute at Chicago

GIF Workshop, Tübingen, May 15-16, 2008

Predicting the next element of a binary sequence

Prediction task

For $t = 1, 2, \dots, T$

- Predict: $\hat{y}_t \in \{\pm 1\}$
- Get: $y_t \in \{\pm 1\}$
- Suffer loss: $\ell_{0-1}(\hat{y}_t, y_t) = \begin{cases} 1 & y_t \neq \hat{y}_t \\ 0 & y_t = \hat{y}_t \end{cases}$

Regret

- Best in hindsight $y^* = \text{sign}(\sum_t y_t)$
- Regret: $R_T = \sum_{t=1}^T \ell_{0-1}(\hat{y}_t, y_t) - \sum_{t=1}^T \ell_{0-1}(y^*, y_t)$

Abstract Prediction Model

- Set of decisions S
- Set of loss functions $\mathcal{L} = \{\ell : S \rightarrow \mathbb{R}\}$

Prediction Game

For $t = 1, \dots, T$

- Learner chooses a decision $\mathbf{w}_t \in S$
 - Environment chooses a loss function $\ell_t \in \mathcal{L}$
 - Learner suffers loss $\ell_t(\mathbf{w}_t)$
-
- **Goal:** Conditions on S and \mathcal{L} that guarantee low regret

$$R_T := \sum_{t=1}^T \ell_t(\mathbf{w}_t) - \sum_{t=1}^T \ell_t(\mathbf{w}^*) \stackrel{!}{=} o(T)$$

- Identifying sufficient conditions for predictability
 - Size matters?
 - No !
 - Maybe yes with randomization ?
 - A modern view: revealing an underlying convexity
- Regret and Convex Duality
- Generality and related work
- Experimental results

Impossibility Result

- $S = \{\pm 1\}$
- $\mathcal{L} = \{\ell_{0-1}(\mathbf{w}_t, 1), \ell_{0-1}(\mathbf{w}_t, -1)\}$
- Adversary can make the cumulative loss of the learner to be T by using $\ell_t(\cdot) = \ell_{0-1}(\cdot, -\mathbf{w}_t)$
- The loss of the constant prediction $w^* = \text{sign}(\sum_t \mathbf{w}_t)$ is at most $T/2$
- Regret is at least $T/2$

Conclusion

- In the above example, $|S| = |\mathcal{L}| = 2$.
- Small size does not guarantee low regret

Solution: Randomized Predictions

- Learner predicts $\hat{y}_t = 1$ with probability w_t
- Best in hindsight: $y_t^* = 1$ with probability w^* where $w^* = \frac{|\{t: y_t=1\}|}{T}$
- Analyze the **expected** regret:

$$\sum_{t=1}^T \mathbb{E}[\hat{y}_t \neq y_t] - \sum_{t=1}^T \mathbb{E}[y_t^* \neq y_t]$$

- There are algorithms that achieve expected regret of $O(\sqrt{T})$

A modern view: revealing an underlying convexity

- Expected zero-one loss can be rewritten as

$$\mathbb{E}[\hat{y}_t \neq y_t] = \begin{cases} 1 - w_t & \text{if } y_t = 1 \\ w_t & \text{if } y_t = -1 \end{cases}$$

- Going back to our abstract model, we get that:
 - $S = [0, 1]$
 - $\mathcal{L} = \{\ell(w) = 1 - w, \ell(w) = w\}$

Properties

- All functions in \mathcal{L} are linear (and thus are convex and Lipschitz)
- S is convex and bounded
- Sufficient conditions for low regret ?

Are we just playing with formalities ?

The convexity assumption is natural in many cases.

Example: Prediction with Expert Advice

- Learner receives a vector $(x_1^t, \dots, x_d^t) \in [-1, 1]^d$ of experts advice
- Learner needs to predict a target $\hat{y}_t \in \mathbb{R}$
- Environment gives correct target $y_t \in \mathbb{R}$
- Learner suffers loss $|y_t - \hat{y}_t|$
- Goal: Be almost as good as the best experts committee

$$\sum_t |y_t - \hat{y}_t| - \sum_t |y_t - \langle \mathbf{w}^*, \mathbf{x}^t \rangle| \stackrel{!}{=} o(T)$$

Are we just playing with formalities ?

The convexity assumption is natural in many cases.

Example: Prediction with Expert Advice

- Learner receives a vector $(x_1^t, \dots, x_d^t) \in [-1, 1]^d$ of experts advice
- Learner needs to predict a target $\hat{y}_t \in \mathbb{R}$
- Environment gives correct target $y_t \in \mathbb{R}$
- Learner suffers loss $|y_t - \hat{y}_t|$
- Goal: Be almost as good as the best experts committee

$$\sum_t |y_t - \hat{y}_t| - \sum_t |y_t - \langle \mathbf{w}^*, \mathbf{x}^t \rangle| \stackrel{!}{=} o(T)$$

Modeling

- S is the d -dimensional probability simplex
- $\mathcal{L} = \{\ell_{\mathbf{x}, y}(\mathbf{w}) = |y - \langle \mathbf{w}, \mathbf{x} \rangle| : \mathbf{x} \in [-1, 1]^d, y \in [-1, 1]\}$

Are we just playing with formalities ?

Example: Convexifying finite decision sets

- Learner should predict an element $s_t \in S' = \{1, \dots, N\}$
- Environment presents non-convex loss function $\ell'_t : S' \rightarrow [0, 1]$
- Learner suffers loss $\ell'_t(s_t)$
- Goal: Be almost as good as the best pure prediction

$$\sum_t \ell'_t(s_t) - \sum_t \ell'_t(s^*) \stackrel{!}{=} o(T)$$

Are we just playing with formalities ?

Example: Convexifying finite decision sets

- Learner should predict an element $s_t \in S' = \{1, \dots, N\}$
- Environment presents non-convex loss function $\ell'_t : S' \rightarrow [0, 1]$
- Learner suffers loss $\ell'_t(s_t)$
- Goal: Be almost as good as the best pure prediction
$$\sum_t \ell'_t(s_t) - \sum_t \ell'_t(s^*) \stackrel{!}{=} o(T)$$

Modeling

- S is the N -dimensional probability simplex
- Prediction s_t is chosen randomly according to $\mathbf{w}_t \in S$
- $\mathcal{L} = \{\ell_{\mathbf{r}}(\mathbf{w}) = \langle \mathbf{w}, \mathbf{r} \rangle : \mathbf{r} \in [0, 1]^N\}$
- $\mathbb{E}[\ell'_t(s_t)] = \ell_{\ell'_t}(\mathbf{w}_t)$

The Online Convex Programming (OCP) model

- All functions in \mathcal{L} are convex and L -Lipschitz
- S is convex and $\max\{\|\mathbf{w}\|_2 : \mathbf{w} \in S\} = D$
- Then, there exists an algorithm with regret $O(LD\sqrt{T})$
- This is tight (i.e. the minimax value of the game)

Bibliography

- The OCP model was presented by Gordon (1999)
- Zinkevich (2003) proved a regret bound of $O((L^2 + D^2)\sqrt{T})$

Dimension independency ?

Yes !

- The regret bound does not depend on the dimensionality of S
- Similarly to Support Vector Machines, we can use Kernel functions

Dimension independency ?

Yes !

- The regret bound does not depend on the dimensionality of S
- Similarly to Support Vector Machines, we can use Kernel functions

Yes ?

- Consider again the prediction with expert advice problem
- d experts, each of which gives an “advice” in $[-1, 1]$
- S is the probability simplex and thus $D = 1$
- Lipschitz constant is $L = \sqrt{d}$
- Regret is $\Omega(\sqrt{dT})$.
- Is this the best we can do ?

Low regret algorithmic framework for OCP

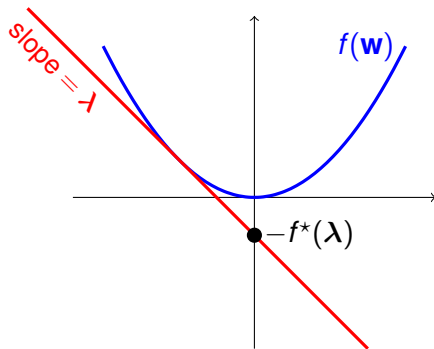
- A low regret algorithmic framework for OCP
- Family of sufficient conditions for low regret
- In particular – Alternatives to the Lipschitz condition
- In the expert committee example – logarithmic dependence on dimension
- Main observation: Relating regret and duality

Fenchel Conjugate

The Fenchel conjugate of the function $f : S \rightarrow \mathbb{R}$ is $f^* : \mathbb{R}^d \rightarrow \mathbb{R}$

$$f^*(\boldsymbol{\lambda}) = \max_{\mathbf{w} \in S} \langle \mathbf{w}, \boldsymbol{\lambda} \rangle - f(\mathbf{w})$$

If f is closed and convex then $f^{**} = f$

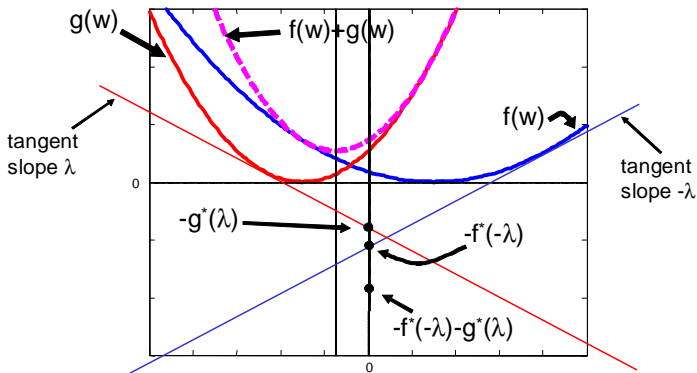


Fenchel Duality

$$\max_{\lambda} -f^*(-\lambda) - g^*(\lambda) \leq \min_{\mathbf{w}} f(\mathbf{w}) + g(\mathbf{w})$$

Fenchel Duality

$$\max_{\lambda} -f^*(-\lambda) - g^*(\lambda) \leq \min_{\mathbf{w}} f(\mathbf{w}) + g(\mathbf{w})$$



Regret and Duality

- Recall that our goal is:

$$\forall \mathbf{w}^* \in \mathcal{S}, \quad \sum_{t=1}^T l_t(\mathbf{w}_t) - \sum_{t=1}^T l_t(\mathbf{w}^*) \leq LD\sqrt{T}$$

Regret and Duality

- Recall that our goal is:

$$\forall \mathbf{w}^* \in \mathcal{S}, \quad \sum_{t=1}^T \ell_t(\mathbf{w}_t) - \sum_{t=1}^T \ell_t(\mathbf{w}^*) \leq LD\sqrt{T}$$

- Rewrite it in a 'silly' way

$$\sum_{t=1}^T \ell_t(\mathbf{w}_t) \leq \min_{\mathbf{w} \in \mathcal{S}} LD\sqrt{T} + \sum_{t=1}^T \ell_t(\mathbf{w})$$

Regret and Duality

- Recall that our goal is:

$$\forall \mathbf{w}^* \in S, \quad \sum_{t=1}^T \ell_t(\mathbf{w}_t) - \sum_{t=1}^T \ell_t(\mathbf{w}^*) \leq LD\sqrt{T}$$

- Rewrite it in a 'silly' way

$$\sum_{t=1}^T \ell_t(\mathbf{w}_t) \leq \min_{\mathbf{w} \in S} LD\sqrt{T} + \sum_{t=1}^T \ell_t(\mathbf{w})$$

- Replace $LD\sqrt{T}$ with a function $f : S \rightarrow \mathbb{R}$ s.t. $\max_{\mathbf{w}} f(\mathbf{w}) \leq LD\sqrt{T}$.
E.g. $f(\mathbf{w}) = c \|\mathbf{w}\|^2$ for $c = L\sqrt{T}/D$. Obtaining:

$$\sum_{t=1}^T \ell_t(\mathbf{w}_t) \leq \min_{\mathbf{w} \in S} f(\mathbf{w}) + \sum_{t=1}^T \ell_t(\mathbf{w})$$

Regret and Duality

- Recall that our goal is:

$$\forall \mathbf{w}^* \in S, \quad \sum_{t=1}^T \ell_t(\mathbf{w}_t) - \sum_{t=1}^T \ell_t(\mathbf{w}^*) \leq LD\sqrt{T}$$

- Rewrite it in a 'silly' way

$$\sum_{t=1}^T \ell_t(\mathbf{w}_t) \leq \min_{\mathbf{w} \in S} LD\sqrt{T} + \sum_{t=1}^T \ell_t(\mathbf{w})$$

- Replace $LD\sqrt{T}$ with a function $f : S \rightarrow \mathbb{R}$ s.t. $\max_{\mathbf{w}} f(\mathbf{w}) \leq LD\sqrt{T}$.
E.g. $f(\mathbf{w}) = c \|\mathbf{w}\|^2$ for $c = L\sqrt{T}/D$. Obtaining:

$$\sum_{t=1}^T \ell_t(\mathbf{w}_t) \leq \min_{\mathbf{w} \in S} f(\mathbf{w}) + \sum_{t=1}^T \ell_t(\mathbf{w})$$

- Lower bound of a minimization problem. **Duality** ?

Properties of the dual problem

$$\max_{\lambda_1, \dots, \lambda_T} -f^*\left(-\sum_t \lambda_t\right) - \sum_t \ell_t^*(\lambda_t) \leq \min_{\mathbf{w} \in S} f(\mathbf{w}) + \sum_{t=1}^T \ell_t(\mathbf{w})$$

Decomposability of the dual

- There's a different dual variable for each online round
- Future loss functions do not affect dual variables of current and past rounds
- Therefore, the dual can be optimized incrementally
- To optimize $\lambda_1, \dots, \lambda_t$, it is enough to know ℓ_1, \dots, ℓ_t

Algorithmic Framework

- Initialize $\lambda_1 = \dots = \lambda_T = \mathbf{0}$
- For $t = 1, 2, \dots, T$
 - Construct \mathbf{w}_t from the dual variables
 - Receive ℓ_t
 - Update dual variables $\lambda_1, \dots, \lambda_t$

Algorithmic Framework

- Initialize $\lambda_1 = \dots = \lambda_T = \mathbf{0}$
- For $t = 1, 2, \dots, T$
 - Construct \mathbf{w}_t from the dual variables
 - Receive ℓ_t
 - Update dual variables $\lambda_1, \dots, \lambda_t$

Lemma

Let \mathcal{D}_t be the dual value at round t and w.l.o.g assume $\mathcal{D}_1 = 0$.

- Assume that $\max_{\mathbf{w} \in \mathcal{S}} f(\mathbf{w}) \leq a\sqrt{T}$
- Assume that $\mathcal{D}_{t+1} - \mathcal{D}_t \geq \ell_t(\mathbf{w}_t) - \frac{a}{\sqrt{T}}$

Then, the regret is bounded by $2a\sqrt{T}$

The proof follows directly from the weak duality theorem

Strong convexity and sufficient dual increase

Strong Convexity w.r.t. norm

A function f is σ -strongly convex over S w.r.t $\| \cdot \|$ if for all $\mathbf{u}, \mathbf{v} \in S$

$$\frac{f(\mathbf{u})+f(\mathbf{v})}{2} \geq f\left(\frac{\mathbf{u}+\mathbf{v}}{2}\right) + \frac{\sigma}{8}\|\mathbf{u} - \mathbf{v}\|^2$$

Lemma (Sufficient Dual Increase)

Assume:

- f is σ -strongly convex w.r.t. $\| \cdot \|$
- ℓ_t is closed and convex
- ∇_t is a sub-gradient of ℓ_t at \mathbf{w}_t

Then, there exists a simple dual update rule s.t.

$$\mathcal{D}_{t+1} - \mathcal{D}_t \geq \ell_t(\mathbf{w}_t) - \frac{\|\nabla_t\|_*^2}{2\sigma}$$

Theorem

Assume:

- *Exists $f : S \rightarrow \mathbb{R}$ which is 1-strongly convex w.r.t. $\| \cdot \|$*
- *$D = \max_{\mathbf{w} \in S} \sqrt{f(\mathbf{w})}$*
- *ℓ_t is closed and convex*
- *$\|\nabla_t\|_* \leq L$ (Lipschitz w.r.t. norm $\| \cdot \|_*$)*

Then, there exists an algorithm with regret bound $2 D L \sqrt{T}$

Generalized Boundedness-Lipschitz condition

Theorem

Assume:

- Exists $f : S \rightarrow \mathbb{R}$ which is 1-strongly convex w.r.t. $\| \cdot \|$
- $D = \max_{\mathbf{w} \in S} \sqrt{f(\mathbf{w})}$
- ℓ_t is closed and convex
- $\|\nabla_t\|_* \leq L$ (Lipschitz w.r.t. norm $\| \cdot \|_*$)

Then, there exists an algorithm with regret bound $2 D L \sqrt{T}$

Example usage – back to expert problem

- Take f to be the relative entropy
- f is strongly convex w.r.t. $\| \cdot \|_1$ and $D = \sqrt{\log(d)}$
- $\|\nabla_t\|_* = \|\mathbf{x}^t\|_\infty \leq 1$
- Regret bound becomes $O(\sqrt{\log(d) T})$

Self Boundedness instead of Lipschitz

Theorem

Replacing Lipschitz condition with the following self-bounded property:

$$\|\nabla_t\| \leq L \sqrt{\ell_t(\mathbf{w}_t)}$$

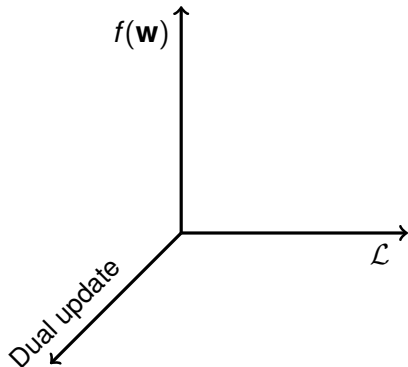
Then,

$$R_T \leq O \left(LD \sqrt{\sum_t \ell_t(\mathbf{w}^*)} + L^2 D^2 \right).$$

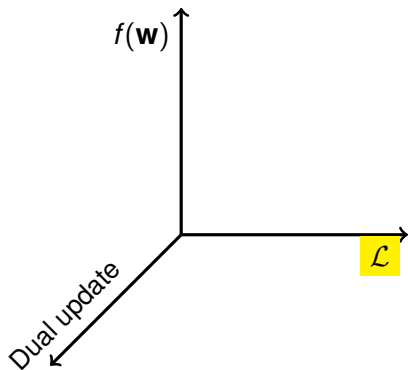
Examples

- $\ell(\mathbf{w}) = \frac{1}{2}(\langle \mathbf{w}, \mathbf{x} \rangle - y)^2$ is $(\sqrt{2}\|\mathbf{x}\|)$ -self-bounded
- $\ell(\mathbf{w}) = \log(1.26 + \exp(-y\langle \mathbf{w}, \mathbf{x} \rangle))$ is $(\|\mathbf{x}\|)$ -self-bounded

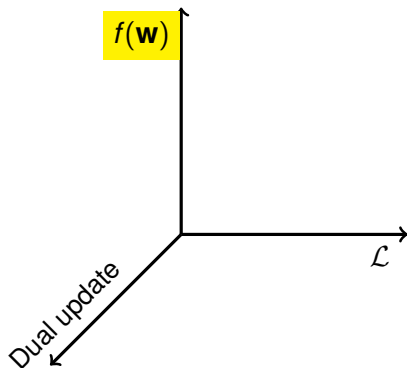
Generality and Related Work



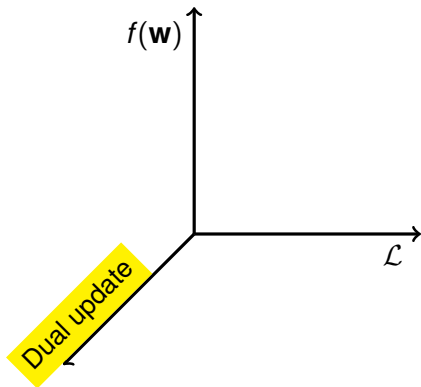
- Family of loss functions (\mathcal{L})



- Online Learning (Perceptron, linear regression, multiclass prediction, structured output, ...)
- Game theory (Playing repeated games, correlated equilibrium)
- Information theory (Prediction of individual sequences)
- Convex optimization (SGD, dual decomposition)



- Complexity function (f)
 - Online learning (Grove, Littlestone, Schuurmans; Kivinen, Warmuth; Gentile; Vovk)
 - Game theory (Hart and Mas-colell)
 - Optimization (Nemirovsky, Yudin; Beck, Teboulle, Nesterov)
 - Unified frameworks (Cesa-Bianchi and Lugosi)

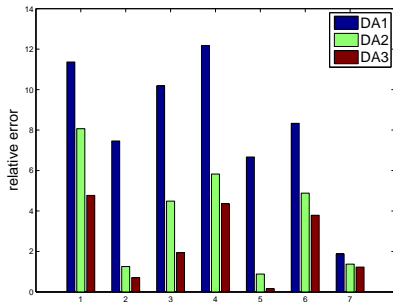


- **Dual update schemes**
 - Only two extremes were studied:
 - Gradient update (naive update of a single dual variable)
 - Follow the leader (Equivalent to full optimization)
 - Our analysis enables the entire spectrum

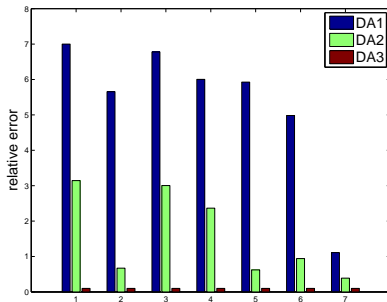
- **Task:** route emails to folders
- 7 users from the **Enron** dataset
- Bag of words representation
- **6 Algorithms**
 - 2 complexity functions (Euclidean and Entropy)
 - 3 dual ascent methods
 - DA1: Fixed sub-gradient ($\lambda_t = s_t \in \partial \ell_t(\mathbf{w}_t)$)
 - DA2: Optimal sub-gradient ($\lambda_t = \alpha_t s_t$ with optimal α_t)
 - DA3: Optimal ($\lambda_t = \arg \max_{\lambda} \mathcal{D}(\lambda_1, \dots, \lambda_{t-1}, \lambda, 0, \dots)$)
- Performance expectation
 - Entropy outperforms Euclidean
 - DA3 better than DA2 better than DA1

Experimental Results – 3 Dual Updates

Euclidean complexity

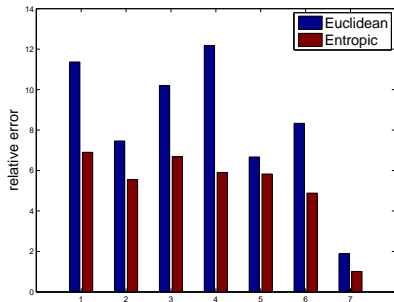


Entropic complexity

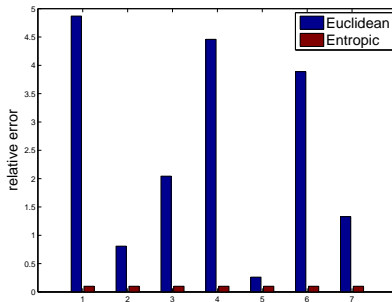


Experimental Results – 2 Complexity Functions

D1



D3



Summary

- The online convex programming is a powerful model
- Achieving low regret by primal-dual algorithmic framework
- Sufficient conditions for predictability

Current and future work

- Logarithmic regret algorithms
- Prediction with limited feedback (Bandit algorithms)
- Boosting, sparsity, and ℓ_1 norm
- Similar sufficient conditions for stochastic optimization (PAC learning)