Mirror Descent

Yoav Freund

February 3, 2025

Material follows Chapter 11 of "Prediction Learning and Games"
Sections 11.{1,2,3}

Outline

Outline

Outline

Linear Pattern Recognition

Potential Based Gradient descent

Duality

Outline

Outline

Linear Pattern Recognition

- Instance: $(\mathbf{x}_t, y_t) \in \mathbb{R}^d \times \mathbb{R}$

Linear Pattern Recognition

- Instance: $(\mathbf{x}_t, y_t) \in \mathbb{R}^d \times \mathbb{R}$
- Expert: $\mathbf{u} \in \mathbb{R}^d$

Linear Pattern Recognition

- Instance: $(\mathbf{x}_t, y_t) \in \mathbb{R}^d \times \mathbb{R}$
- Expert: $\mathbf{u} \in \mathbb{R}^d$
- Predictor: $\mathbf{w}_t \in \mathbb{R}^d$

Linear Pattern Recognition

- Instance: $(\mathbf{x}_t, y_t) \in \mathbb{R}^d \times \mathbb{R}$
- Expert: $\mathbf{u} \in \mathbb{R}^d$
- Predictor: $\mathbf{w}_t \in \mathbb{R}^d$
- Loss $\ell(\mathbf{w} \cdot \mathbf{x}, y)$ (online regression = square loss)

Linear Pattern Recognition

- Instance: $(\mathbf{x}_t, y_t) \in \mathbb{R}^d \times \mathbb{R}$
- Expert: $\mathbf{u} \in \mathbb{R}^d$
- Predictor: $\mathbf{w}_t \in \mathbb{R}^d$
- Loss $\ell(\mathbf{w} \cdot \mathbf{x}, y)$ (online regression = square loss)
- Regret: $\mathbf{R}_t(\mathbf{u}) = \sum_{i=1}^{t} [\ell(\mathbf{w}_t \cdot \mathbf{x}_t, y_t) - \ell(\mathbf{u} \cdot \mathbf{x}_t, y_t)]$

Potential based gradient Descent

- $\mathbf{R}_t$ = Regret vector $R_t(\mathbf{w}) = L_{A,t} - L_t(\mathbf{w})$

Potential based gradient Descent

- $\mathbf{R}_t$ = Regret vector $R_t(\mathbf{w}) = L_{A,t} - L_t(\mathbf{w})$
- $\mathbf{R}_t$ = State of prediction algorithm at time t

Potential based gradient Descent

- $\mathbf{R}_t$ = Regret vector $R_t(\mathbf{w}) = L_{A,t} - L_t(\mathbf{w})$
- $\mathbf{R}_t$ = State of prediction algorithm at time t
- Potential: $\Phi(\mathbf{R})$ Quantifies badness of the state.

Potential based gradient Descent

- $\mathbf{R}_t$ = Regret vector $R_t(\mathbf{w}) = L_{A,t} - L_t(\mathbf{w})$
- $\mathbf{R}_t$ = State of prediction algorithm at time t
- Potential: $\Phi(\mathbf{R})$ Quantifies badness of the state.
- A state is bad if adversary can force high regret in the future.

Potential based gradient Descent

- $\mathbf{R}_t$ = Regret vector $R_t(\mathbf{w}) = L_{A,t} - L_t(\mathbf{w})$
- $\mathbf{R}_t$ = State of prediction algorithm at time t
- Potential: $\Phi(\mathbf{R})$ Quantifies badness of the state.
- A state is bad if adversary can force high regret in the future.
- Choose prediction to balance $\Phi(\mathbf{R}_{t+1}) - \Phi(\mathbf{R}_t) + \mathbf{w}_t \cdot \ell_t$ is small for all possible ℓ_t

Potential based gradient Descent

- $\mathbf{R}_t$ = Regret vector $R_t(\mathbf{w}) = L_{A,t} - L_t(\mathbf{w})$
- $\mathbf{R}_t$ = State of prediction algorithm at time t
- Potential: $\Phi(\mathbf{R})$ Quantifies badness of the state.
- A state is bad if adversary can force high regret in the future.
- Choose prediction to balance $\Phi(\mathbf{R}_{t+1}) - \Phi(\mathbf{R}_t) + \mathbf{w}_t \cdot \ell_t$ is small for all possible ℓ_t
- $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$ is a good choice.

Potential based gradient Descent

- $\mathbf{R}_t$ = Regret vector $R_t(\mathbf{w}) = L_{A,t} - L_t(\mathbf{w})$
- $\mathbf{R}_t$ = State of prediction algorithm at time t
- Potential: $\Phi(\mathbf{R})$ Quantifies badness of the state.
- A state is bad if adversary can force high regret in the future.
- Choose prediction to balance $\Phi(\mathbf{R}_{t+1}) - \Phi(\mathbf{R}_t) + \mathbf{w}_t \cdot \ell_t$ is small for all possible ℓ_t
- $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$ is a good choice.
- For finite number of experts, $\mathbf{R}_t$ is finite dimensional and we can compute $\mathbf{w}_t$ explicitly.

Potential based gradient Descent

- $\mathbf{R}_t$ = Regret vector $R_t(\mathbf{w}) = L_{A,t} - L_t(\mathbf{w})$
- $\mathbf{R}_t$ = State of prediction algorithm at time t
- Potential: $\Phi(\mathbf{R})$ Quantifies badness of the state.
- A state is bad if adversary can force high regret in the future.
- Choose prediction to balance $\Phi(\mathbf{R}_{t+1}) - \Phi(\mathbf{R}_t) + \mathbf{w}_t \cdot \ell_t$ is small for all possible ℓ_t
- $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$ is a good choice.
- For finite number of experts, $\mathbf{R}_t$ is finite dimensional and we can compute $\mathbf{w}_t$ explicitly.
- Here, $\mathbf{R} = \{R(\mathbf{w})\}_{\mathbf{w}\in\mathbb{R}^d}$ is continuous dimensional.

Potential based gradient Descent

- $\mathbf{R}_t$ = Regret vector $R_t(\mathbf{w}) = L_{A,t} - L_t(\mathbf{w})$
- $\mathbf{R}_t$ = State of prediction algorithm at time t
- Potential: $\Phi(\mathbf{R})$ Quantifies badness of the state.
- A state is bad if adversary can force high regret in the future.
- Choose prediction to balance $\Phi(\mathbf{R}_{t+1}) - \Phi(\mathbf{R}_t) + \mathbf{w}_t \cdot \ell_t$ is small for all possible ℓ_t
- $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$ is a good choice.
- For finite number of experts, $\mathbf{R}_t$ is finite dimensional and we can compute $\mathbf{w}_t$ explicitly.
- Here, $\mathbf{R} = \{R(\mathbf{w})\}_{\mathbf{w}\in\mathbb{R}^d}$ is continuous dimensional.
- Experts that correspond to exponential distributions - we can use conjugate priors. (recall: biased coins).

Potential based gradient Descent

- $\mathbf{R}_t$ = Regret vector $R_t(\mathbf{w}) = L_{A,t} - L_t(\mathbf{w})$
- $\mathbf{R}_t$ = State of prediction algorithm at time t
- Potential: $\Phi(\mathbf{R})$ Quantifies badness of the state.
- A state is bad if adversary can force high regret in the future.
- Choose prediction to balance $\Phi(\mathbf{R}_{t+1}) - \Phi(\mathbf{R}_t) + \mathbf{w}_t \cdot \ell_t$ is small for all possible ℓ_t
- $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$ is a good choice.
- For finite number of experts, $\mathbf{R}_t$ is finite dimensional and we can compute $\mathbf{w}_t$ explicitly.
- Here, $\mathbf{R} = \{R(\mathbf{w})\}_{\mathbf{w}\in\mathbb{R}^d}$ is continuous dimensional.
- Experts that correspond to exponential distributions - we can use conjugate priors. (recall: biased coins).
- We need a new trick to compute $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$ efficiently.

Dual Vector Spaces

- V is a vector space, with a norm $\|v\|$

Dual Vector Spaces

- V is a vector space, with a norm $\|v\|$
- U is the set of all linear mappings from V to V

Dual Vector Spaces

- V is a vector space, with a norm $\|v\|$
- U is the set of all linear mappings from V to V
- The norm of $u \in U$ is defined as

$$\|u\|^* = \max_{v \in V} \frac{\|u(v)\|}{\|v\|}$$

Dual Vector Spaces

- V is a vector space, with a norm $\|v\|$
- U is the set of all linear mappings from V to V
- The norm of $u \in U$ is defined as

$$\|u\|^* = \max_{v \in V} \frac{\|u(v)\|}{\|v\|}$$

- V is equivalent to the set of all linear mappings from U to U.

Dual Vector Spaces

- V is a vector space, with a norm $\|v\|$
- U is the set of all linear mappings from V to V
- The norm of $u \in U$ is defined as

$$\|u\|^* = \max_{v \in V} \frac{\|u(v)\|}{\|v\|}$$

- V is equivalent to the set of all linear mappings from U to U.
- U and V are dual vector spaces, with dual norms.

Dual Norms

- this

Dual Norms

- this
- The space is always $U, V = \mathbb{R}^n$

Dual Norms

- this
- The space is always $U, V = \mathbb{R}^n$
- The linear operation is the dot product $\mathbf{u} \cdot \mathbf{v}$

Dual Norms

- this
- The space is always $U, V = \mathbb{R}^n$
- The linear operation is the dot product $\mathbf{u} \cdot \mathbf{v}$
- L_2 norm: $\sqrt{\sum_{i=1}^{n} x_i^2}$

Dual Norms

- this
- The space is always $U, V = \mathbb{R}^n$
- The linear operation is the dot product $\mathbf{u} \cdot \mathbf{v}$
- L_2 norm: $\sqrt{\sum_{i=1}^{n} x_i^2}$
- L_1 norm: $\sum_{i=1}^{n} |x_i|$

Dual Norms

- this
- The space is always $U, V = \mathbb{R}^n$
- The linear operation is the dot product $\mathbf{u} \cdot \mathbf{v}$
- L_2 norm: $\sqrt{\sum_{i=1}^n x_i^2}$
- L_1 norm: $\sum_{i=1}^n |x_i|$
- L_∞ norm: $\max_i |x_i|$

Dual Norms

- this
- The space is always $U, V = \mathbb{R}^n$
- The linear operation is the dot product $\mathbf{u} \cdot \mathbf{v}$
- L_2 norm: $\sqrt{\sum_{i=1}^{n} x_i^2}$
- L_1 norm: $\sum_{i=1}^{n} |x_i|$
- L_∞ norm: $\max_i |x_i|$
- L_p norm: $\left(\sum_{i=1}^{n} x_i^p\right)^{\frac{1}{p}}$

Dual Norms

- this
- The space is always $U, V = \mathbb{R}^n$
- The linear operation is the dot product $\mathbf{u} \cdot \mathbf{v}$
- L_2 norm: $\sqrt{\sum_{i=1}^n x_i^2}$
- L_1 norm: $\sum_{i=1}^n |x_i|$
- L_∞ norm: $\max_i |x_i|$
- L_p norm: $\left(\sum_{i=1}^n x_i^p\right)^{\frac{1}{p}}$
- L_p, L_q are dual norms if $p, q \geq 1, \text{ and } \frac{1}{p} + \frac{1}{q} = 1$

Dual Norms

- this
- The space is always $U, V = \mathbb{R}^n$
- The linear operation is the dot product $\mathbf{u} \cdot \mathbf{v}$
- L_2 norm: $\sqrt{\sum_{i=1}^n x_i^2}$
- L_1 norm: $\sum_{i=1}^n |x_i|$
- L_∞ norm: $\max_i |x_i|$
- L_p norm: $\left(\sum_{i=1}^n x_i^p\right)^{\frac{1}{p}}$
- L_p, L_q are dual norms if $p, q \geq 1, \text{ and } \frac{1}{p} + \frac{1}{q} = 1$
- L_1, L_∞ are dual.

Dual Norms

- this
- The space is always $U, V = \mathbb{R}^n$
- The linear operation is the dot product $\mathbf{u} \cdot \mathbf{v}$
- L_2 norm: $\sqrt{\sum_{i=1}^{n} x_i^2}$
- L_1 norm: $\sum_{i=1}^{n} |x_i|$
- L_∞ norm: $\max_i |x_i|$
- L_p norm: $\left(\sum_{i=1}^{n} x_i^p\right)^{\frac{1}{p}}$
- L_p, L_q are dual norms if $p, q \geq 1, \text{ and } \frac{1}{p} + \frac{1}{q} = 1$
- L_1, L_∞ are dual.
- L_2 is self-dual.

Fenchel Duality

- Fenchel duality allows non-differentiable functions and uses sub-gradient

Fenchel Duality

- Fenchel duality allows non-differentiable functions and uses sub-gradient
- Suppose $F : A \to \mathbb{R}$ is a convex function over a convex set $A \subseteq \mathbb{R}^n$.

Fenchel Duality

- Fenchel duality allows non-differentiable functions and uses sub-gradient
- Suppose $F : A \to \mathbb{R}$ is a convex function over a convex set $A \subseteq \mathbb{R}^n$.
- The dual function to F is

$$F^*(\mathbf{u}) = \sup_{\mathbf{v} \in A} (\mathbf{u} \cdot \mathbf{v} - F(\mathbf{v}))$$

Visualization for $\mathbb{R}$

Visualization for $\mathbb{R}$

Visualization for $\mathbb{R}$

- $x, y\mathbb{R}$

Visualization for $\mathbb{R}$

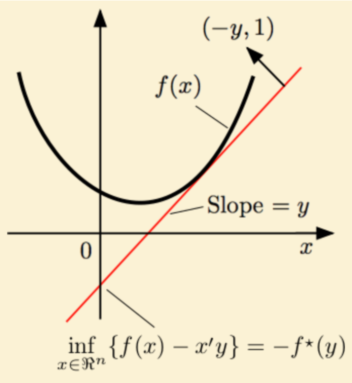
- $x, y\mathbb{R}$
- $f^*(y) = \sup_{x\in\mathbb{R}} (xy - f(x))$

Visualization for $\mathbb{R}$

- $x, y\mathbb{R}$
- $f^*(y) = \sup_{x \in \mathbb{R}} (xy - f(x))$
- $-f^*(y) = \inf_{x \in \mathbb{R}} (f(x) - xy)$

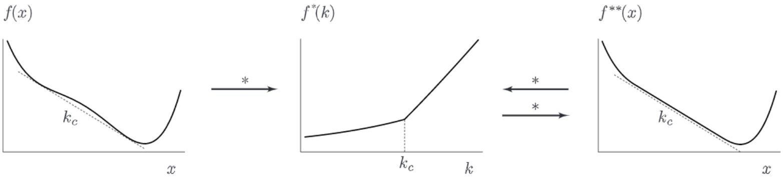
Visualization for $\mathbb{R}$

- $x, y\mathbb{R}$
- $f^*(y) = \sup_{x\in\mathbb{R}} (xy - f(x))$
- $-f^*(y) = \inf_{x\in\mathbb{R}} (f(x) - xy)$



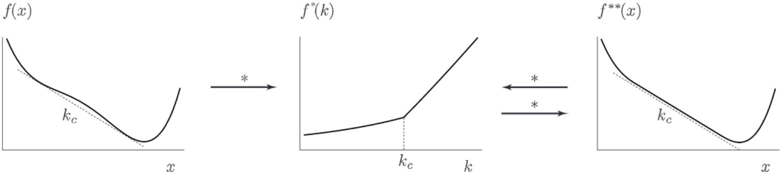
Dual of Dual

- The dual of any function is convex.



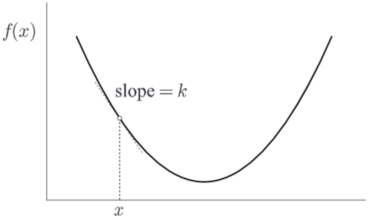
Dual of Dual

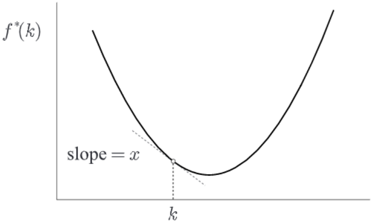
- The dual of any function is convex.
- if F is convex then $F^{**} = F$



Gradient Duality

- If the gradient of f at x is k then the gradient of f^* at k is x

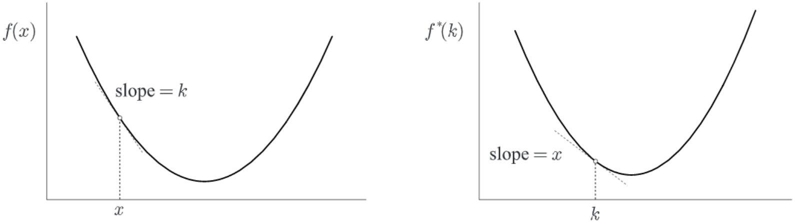




Gradient Duality

- If the gradient of f at x is k then the gradient of f^* at k is x
- In general:

$$\nabla F^* = (\nabla F)^{-1}$$



Example: Exponential Potential

- Potential: $F(\mathbf{u}) = \sum_{i=1}^{d} e^{u_i}$

Example: Exponential Potential

- Potential: $F(\mathbf{u}) = \sum_{i=1}^{d} e^{u_i}$
- Gradient: $\nabla F(\mathbf{u})_i = e^{u_i}$ or $\nabla F(\mathbf{u}) = F(\mathbf{u})$.

Example: Exponential Potential

- Potential: $F(\mathbf{u}) = \sum_{i=1}^{d} e^{u_i}$
- Gradient: $\nabla F(\mathbf{u})_i = e^{u_i}$ or $\nabla F(\mathbf{u}) = F(\mathbf{u})$.
- Dual: $F^*(\mathbf{v}) = \sum_{i=1}^{d} v_i(\ln v_i - 1)$

Example: Exponential Potential

- Potential: $F(\mathbf{u}) = \sum_{i=1}^{d} e^{u_i}$
- Gradient: $\nabla F(\mathbf{u})_i = e^{u_i}$ or $\nabla F(\mathbf{u}) = F(\mathbf{u})$.
- Dual: $F^*(\mathbf{v}) = \sum_{i=1}^{d} v_i(\ln v_i - 1)$
- Gradient of dual: $\nabla F^*(\mathbf{v})_i = \ln v_i$

Example: Exponential Potential

- Potential: $F(\mathbf{u}) = \sum_{i=1}^{d} e^{u_i}$
- Gradient: $\nabla F(\mathbf{u})_i = e^{u_i}$ or $\nabla F(\mathbf{u}) = F(\mathbf{u})$.
- Dual: $F^*(\mathbf{v}) = \sum_{i=1}^{d} v_i(\ln v_i - 1)$
- Gradient of dual: $\nabla F^*(\mathbf{v})_i = \ln v_i$
- Note $(\nabla F)^{-1} = \nabla F^*$

Fenchel and Bregman

- F: strictly convex with continuous first derivative.

Fenchel and Bregman

- F: strictly convex with continuous first derivative.
- F^* is the Fenchel Dual of F

Fenchel and Bregman

- F: strictly convex with continuous first derivative.
- F^* is the Fenchel Dual of F
- D_F, D_{F^*} Bregman divergences wrt F, F^*

Fenchel and Bregman

- F: strictly convex with continuous first derivative.
- F^* is the Fenchel Dual of F
- D_F, D_{F^*} Bregman divergences wrt F, F^*
- $\mathbf{u}' = \nabla F(\mathbf{u})$ and $\mathbf{v}' = \nabla F(\mathbf{v})$

Fenchel and Bregman

- F: strictly convex with continuous first derivative.
- F^* is the Fenchel Dual of F
- D_F, D_{F^*} Bregman divergences wrt F, F^*
- $\mathbf{u}' = \nabla F(\mathbf{u})$ and $\mathbf{v}' = \nabla F(\mathbf{v})$
- $D_F(\mathbf{u}, \mathbf{v}) = D_{F^*}(\mathbf{u}', \mathbf{v}')$

Dual parameters

- We want to compute $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$

Dual parameters

- We want to compute $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$
- Let Φ^* by the convex Dual of Φ

Dual parameters

- We want to compute $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$
- Let Φ^* by the convex Dual of Φ
- $\mathbf{R}_t = \nabla\Phi^*(\mathbf{w}_t)$

Dual parameters

- We want to compute $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$
- Let Φ^* by the convex Dual of Φ
- $\mathbf{R}_t = \nabla\Phi^*(\mathbf{w}_t)$
- We use $\theta_t = \mathbf{R}_t$ because we treat $\mathbf{R}_t$ as a parameter.

Dual parameters

- We want to compute $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$
- Let Φ^* by the convex Dual of Φ
- $\mathbf{R}_t = \nabla\Phi^*(\mathbf{w}_t)$
- We use $\theta_t = \mathbf{R}_t$ because we treat $\mathbf{R}_t$ as a parameter.
- $\mathbf{r}_t$ regret for single step.

Dual parameters

- We want to compute $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$
- Let Φ^* by the convex Dual of Φ
- $\mathbf{R}_t = \nabla\Phi^*(\mathbf{w}_t)$
- We use $\theta_t = \mathbf{R}_t$ because we treat $\mathbf{R}_t$ as a parameter.
- $\mathbf{r}_t$ regret for single step.
- $\theta_t = \theta_{t-1} + \mathbf{r}_t$

Dual parameters

- We want to compute $\mathbf{w}_t = \nabla\Phi(\mathbf{R}_t)$
- Let Φ^* by the convex Dual of Φ
- $\mathbf{R}_t = \nabla\Phi^*(\mathbf{w}_t)$
- We use $\theta_t = \mathbf{R}_t$ because we treat $\mathbf{R}_t$ as a parameter.
- $\mathbf{r}_t$ regret for single step.
- $\theta_t = \theta_{t-1} + \mathbf{r}_t$
- re-written using Duality:

$$\nabla\Phi^*(\mathbf{w}_t) = \nabla\Phi(\mathbf{w}_{t-1}) + \mathbf{r}_t$$

Mirror Descent

- Gradient descent in dual space $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \lambda \nabla \ell_t(\boldsymbol{\theta}_{t-1})$

Mirror Descent

- Gradient descent in dual space $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \lambda \nabla \ell_t(\boldsymbol{\theta}_{t-1})$
- Using duality can be rewritten as

$$\nabla \Phi^*(\mathbf{w}_t) = \nabla \Phi^*(\mathbf{w}_{t-1}) - \lambda \nabla \ell_t(\mathbf{w}_{t-1})$$

Mirror Descent

- Gradient descent in dual space $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \lambda \nabla \ell_t(\boldsymbol{\theta}_{t-1})$
- Using duality can be rewritten as

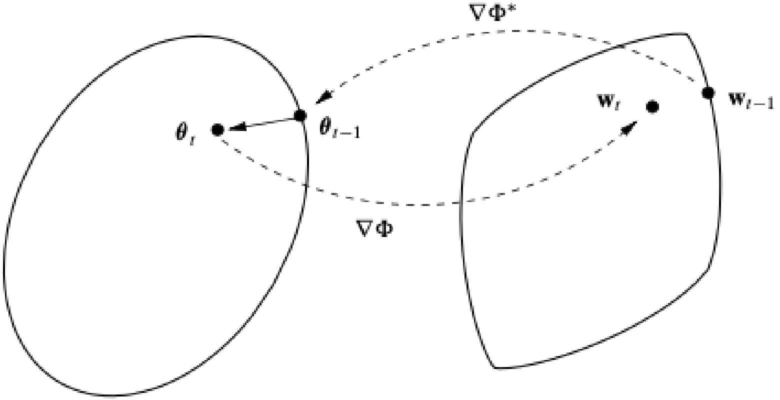
$$\nabla \Phi^*(\mathbf{w}_t) = \nabla \Phi^*(\mathbf{w}_{t-1}) - \lambda \nabla \ell_t(\mathbf{w}_{t-1})$$

- As $\nabla \Phi$ is the inverse of $\nabla \Phi^*$ we get

$$\mathbf{w}_t = \nabla \Phi(\nabla \Phi^*(\mathbf{w}_{t-1}) - \lambda \nabla \ell_t(\mathbf{w}_{t-1}))$$

A picture of mirror descent

$$\mathbf{w}_t = \nabla\Phi(\nabla\Phi^*(\mathbf{w}_{t-1}) - \lambda\nabla\ell_t(\mathbf{w}_{t-1}))$$



Intuition

- $\mathbf{u}$ should balance minimizing the loss from observing same example again and divergence between $\mathbf{u}$ and $\mathbf{w}_{t-1}$

Intuition

- $\mathbf{u}$ should balance minimizing the loss from observing same example again and divergence between $\mathbf{u}$ and $\mathbf{w}_{t-1}$
- Exact Goal: $\min_{\mathbf{u} \in \mathbb{R}^d} [D_{\phi^*}(\mathbf{u}, \mathbf{w}_{t-1}) - \lambda \nabla \ell_t(\mathbf{u})]$

Intuition

- $\mathbf{u}$ should balance minimizing the loss from observing same example again and divergence between $\mathbf{u}$ and $\mathbf{w}_{t-1}$
- Exact Goal: $\min_{\mathbf{u}\in\mathbb{R}^d} [D_{\phi^*}(\mathbf{u}, \mathbf{w}_{t-1}) - \lambda\nabla\ell_t(\mathbf{u})]$
- Taylor order one approximation: $\min_{\mathbf{u}\in\mathbb{R}^d} [F(\mathbf{u})]$ where $F(\mathbf{u}) = D_{\phi^*}(\mathbf{u}, \mathbf{w}_{t-1}) - \lambda[\ell_t(\mathbf{w}_{t-1}) + (\mathbf{u} - \mathbf{w}_{t-1})\nabla\ell_t(\mathbf{w}_{t-1})]$

Intuition

- $\mathbf{u}$ should balance minimizing the loss from observing same example again and divergence between $\mathbf{u}$ and $\mathbf{w}_{t-1}$
- Exact Goal: $\min_{\mathbf{u}\in\mathbb{R}^d} [D_{\phi^*}(\mathbf{u},\mathbf{w}_{t-1}) - \lambda\nabla\ell_t(\mathbf{u})]$
- Taylor order one approximation: $\min_{\mathbf{u}\in\mathbb{R}^d} [F(\mathbf{u})]$ where $F(\mathbf{u}) = D_{\phi^*}(\mathbf{u},\mathbf{w}_{t-1}) - \lambda[\ell_t(\mathbf{w}_{t-1}) + (\mathbf{u}-\mathbf{w}_{t-1})\nabla\ell_t(\mathbf{w}_{t-1})]$
- Assuming everything is differrentiable and convex, $\nabla_{\mathbf{u}} F[\mathbf{u}] = 0$ yields: $\nabla\Phi^*(\mathbf{w}_t) = \nabla\Phi^*(\mathbf{w}_{t-1}) - \lambda\nabla\ell_t(\mathbf{w}_{t-1})$

Intuition

- $\mathbf{u}$ should balance minimizing the loss from observing same example again and divergence between $\mathbf{u}$ and $\mathbf{w}_{t-1}$
- Exact Goal: $\min_{\mathbf{u}\in\mathbb{R}^d}[D_{\phi^*}(\mathbf{u},\mathbf{w}_{t-1}) - \lambda\nabla\ell_t(\mathbf{u})]$
- Taylor order one approximation: $\min_{\mathbf{u}\in\mathbb{R}^d}[F(\mathbf{u})]$ where $F(\mathbf{u}) = D_{\phi^*}(\mathbf{u},\mathbf{w}_{t-1}) - \lambda[\ell_t(\mathbf{w}_{t-1}) + (\mathbf{u}-\mathbf{w}_{t-1})\nabla\ell_t(\mathbf{w}_{t-1})]$
- Assuming everything is differrentiable and convex, $\nabla_{\mathbf{u}}F[\mathbf{u}] = 0$ yields: $\nabla\Phi^*(\mathbf{w}_t) = \nabla\Phi^*(\mathbf{w}_{t-1}) - \lambda\nabla\ell_t(\mathbf{w}_{t-1})$
- Equivelently: $\mathbf{w}_t = \nabla\Phi(\nabla\Phi^*(\mathbf{w}_{t-1}) - \lambda\nabla\ell_t(\mathbf{w}_{t-1}))$

Theorem

- $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a regular loss function if it is convex, non-negative and differentiable.

Theorem

- $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a regular loss function if it is convex, non-negative and differentiable.
- Instantaneous Loss: $\ell_t(\mathbf{w}) = \ell(\mathbf{w} \cdot \mathbf{x}_t, y_t)$

Theorem

- $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a regular loss function if it is convex, non-negative and differentiable.
- Instantaneous Loss: $\ell_t(\mathbf{w}) = \ell(\mathbf{w} \cdot \mathbf{x}_t, y_t)$
- Regret: $\mathbf{R}_t(\mathbf{u}) = L_{A,t} - L_t(\mathbf{u})$

Theorem

- $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a regular loss function if it is convex, non-negative and differentiable.
- Instantaneous Loss: $\ell_t(\mathbf{w}) = \ell(\mathbf{w} \cdot \mathbf{x}_t, y_t)$
- Regret: $\mathbf{R}_t(\mathbf{u}) = L_{A,t} - L_t(\mathbf{u})$
- Theorem: For all example sequences $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$, any initial vector $\mathbf{w}_0 \in \mathbb{R}^d$. all $\lambda > 0$ and all $\mathbf{u} \in \mathbb{R}^d$:

$$\mathbf{R}_T(\mathbf{u}) \leq \frac{1}{\lambda} D_{\Phi^*}(\mathbf{u}, \mathbf{w}_0) + \frac{1}{\lambda} \sum_{t=1}^{T} D_{\Phi^*}(\mathbf{w}_{t-1}, \mathbf{w}_t)$$

Polynomial Potential

- Potential: $\Phi_p(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|_p^2 = \frac{1}{2}\left(\sum_{i=1}^d u_i^p\right)^{2/p}$

Polynomial Potential

- Potential: $\Phi_p(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|_p^2 = \frac{1}{2}\left(\sum_{i=1}^d u_i^p\right)^{2/p}$
- Dual Potential $\Phi_p^* = \Phi_q$ Where $\frac{1}{p} + \frac{1}{q} = 1$

Polynomial Potential

- Potential: $\Phi_p(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|_p^2 = \frac{1}{2}\left(\sum_{i=1}^{d} u_i^p\right)^{2/p}$
- Dual Potential $\Phi_p^* = \Phi_q$ Where $\frac{1}{p} + \frac{1}{q} = 1$
- Euclidean norm: $q = p = 2$

Polynomial Potential

- Potential: $\Phi_p(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|_p^2 = \frac{1}{2}\left(\sum_{i=1}^d u_i^p\right)^{2/p}$
- Dual Potential $\Phi_p^* = \Phi_q$ Where $\frac{1}{p} + \frac{1}{q} = 1$
- Euclidean norm: $q = p = 2$
- Suppose the sequence of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ satisfies $\|\mathbf{x}_t\|_p \leq X_p$ for all $1 \leq t \leq T$

Polynomial Potential

- Potential: $\Phi_p(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|_p^2 = \frac{1}{2}\left(\sum_{i=1}^{d} u_i^p\right)^{2/p}$
- Dual Potential $\Phi_p^* = \Phi_q$ Where $\frac{1}{p} + \frac{1}{q} = 1$
- Euclidean norm: $q = p = 2$
- Suppose the sequence of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ satisfies $\|\mathbf{x}_t\|_p \leq X_p$ for all $1 \leq t \leq T$
- Suppose we use the dual descend algorithm for the potential function Φ_p and the learning rate $\lambda = \frac{2\epsilon}{(p-1)X_p^2}$ for some $0 < \epsilon < 1$

Polynomial Potential

- Potential: $\Phi_p(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|_p^2 = \frac{1}{2}\left(\sum_{i=1}^d u_i^p\right)^{2/p}$
- Dual Potential $\Phi_p^* = \Phi_q$ Where $\frac{1}{p} + \frac{1}{q} = 1$
- Euclidean norm: $q = p = 2$
- Suppose the sequence of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ satisfies $\|\mathbf{x}_t\|_p \leq X_p$ for all $1 \leq t \leq T$
- Suppose we use the dual descend algorithm for the potential function Φ_p and the learning rate $\lambda = \frac{2\epsilon}{(p-1)X_p^2}$ for some $0 < \epsilon < 1$
- Loss Bound:
  $L_{A,T} \leq \frac{L_T(\mathbf{u})}{1-\epsilon} + \frac{\|\mathbf{u}\|_q^2}{\epsilon(1-\epsilon)} \times \frac{(p-1)X_p^2}{4}$

Exponential Potential

- Potential: $\Phi(\mathbf{u}) = \sum_{i=1}^{d} e^{u_i}$

Exponential Potential

- Potential: $\Phi(\mathbf{u}) = \sum_{i=1}^{d} e^{u_i}$
- Dual Potential $\Phi^*(\mathbf{u}) = \sum_{i=1}^{d} u_i(\ln u_i - 1)$

Exponential Potential

- Potential: $\Phi(\mathbf{u}) = \sum_{i=1}^{d} e^{u_i}$
- Dual Potential $\Phi^*(\mathbf{u}) = \sum_{i=1}^{d} u_i(\ln u_i - 1)$
- Euclidean norm: $q = p = 2$

Exponential Potential

- Potential: $\Phi(\mathbf{u}) = \sum_{i=1}^{d} e^{u_i}$
- Dual Potential $\Phi^*(\mathbf{u}) = \sum_{i=1}^{d} u_i(\ln u_i - 1)$
- Euclidean norm: $q = p = 2$
- Suppose the sequence of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ satisfies $\|\mathbf{x}_t\|_\infty \leq X_p$ for all $1 \leq t \leq T$

Exponential Potential

- Potential: $\Phi(\mathbf{u}) = \sum_{i=1}^{d} e^{u_i}$
- Dual Potential $\Phi^*(\mathbf{u}) = \sum_{i=1}^{d} u_i(\ln u_i - 1)$
- Euclidean norm: $q = p = 2$
- Suppose the sequence of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ satisfies $\|\mathbf{x}_t\|_\infty \leq X_p$ for all $1 \leq t \leq T$
- Suppose we use the dual descend algorithm for the exponential potential function Φ and the learning rate $\lambda = \frac{2\epsilon}{X_\infty^2}$ for some $0 < \epsilon < 1$

Exponential Potential

- Potential: $\Phi(\mathbf{u}) = \sum_{i=1}^{d} e^{u_i}$
- Dual Potential $\Phi^*(\mathbf{u}) = \sum_{i=1}^{d} u_i(\ln u_i - 1)$
- Euclidean norm: $q = p = 2$
- Suppose the sequence of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ satisfies $\|\mathbf{x}_t\|_\infty \leq X_p$ for all $1 \leq t \leq T$
- Suppose we use the dual descend algorithm for the exponential potential function Φ and the learning rate $\lambda = \frac{2\epsilon}{X_\infty^2}$ for some $0 < \epsilon < 1$
- Loss Bound:
  $L_{A,T} \leq \frac{L_T(\mathbf{u})}{1-\epsilon} + \frac{X_\infty^2 \ln d}{2\epsilon(1-\epsilon)}$