Evaluation

Online learning

links with optimization and games

Université Paris–Saclay

adagrad on the simplex and application to games

Let $d \geqslant 1$ and consider the simplex $\mathcal{X} = \Delta_d$.

1) Derive regret bounds for AdaGrad-Norm and AdaGrad-Diagonal in the special case of the simplex.

2) Let $m, n \geqslant 1$ be integers and $A \in \mathbb{R}^{m \times n}$. In the context of solving the two-player zero-sum game given by A, derive guarantees for AdaGrad-Norm and AdaGrad-Diagonal.

3) Implement AdaGrad-Norm on the simplex with the help of the following snippet which computes the Euclidean projection onto the simplex.

```
def projection_simplex(y):
    n_features = y.shape[0]
    z = np.sort(y)[::-1]
    cssv = np.cumsum(z) - 1
    ind = np.arange(n_features) + 1
    cond = u - cssv / ind > 0
    rho = ind[cond][-1]
    theta = cssv[cond][-1] / float(rho)
```

```python
    w = np.maximum(y - theta, 0)
    return w
```

4) In the context of solving two-player zero-sum games, perform numerical experiments to compare the performance of AdaGrad-Norm with RM, RM+ and the exponential weights algorithm. Also include optimistic variants of all algorithms.

5) Bonus. — Rewrite the above Python function mathematically and prove that it indeed computes the Euclidean projection onto the simplex.

6) Bonus. — Also implement AdaGrad-Diagonal and add it to the numerical experiments.