Online Convex Optimization

- **OCO**: in each step the learner presents its prediction (hypothesis) \(u_t\) and then the adversary picks a loss function \(f_t(\cdot)\).
- The constraints are that \(u_t \in S\) a convex set, and that \(f_t\) must be convex.
- Typically \(f_t\) is given via \(z_t = (x_t, y_t)\) where \(f_t(u) = \ell(u, (x_t, y_t))\) for example using linear separators and the 0-1 loss \(\ell(u; (x_t, y_t)) = [\text{sign} (u^T x_t) \neq y_t]\) but this is not necessary for the development.
- The Regret of the learner with respect to a fixed predictor \(u\) is
  \[
  R_T = \sum_{t=1}^{T} f_t(u_t) - \sum_{t=1}^{T} f_t(u).
  \]
  The regret of the learner is the regret against the best \(u\).

- **OCO for Risk Minimization**: \(\text{Risk}(u) = E_z \ell(u, z)\)
  - run OCO algorithm (e.g., OPGD) using sequence of \(f_t(u_t)\)
  - Here \(f_t(u_t)\) induced by IID \(z_t\) choices
  - Then w.p. \(\geq 1 - \delta\) for \(\hat{w} = \frac{1}{T} \sum_{i=1}^{T} w_i\)
  \[
  \text{Risk}(\hat{w}) \leq \text{Risk}(u) + \frac{1}{T} R_T + 2 \sqrt{\frac{2}{T} \ln \frac{1}{\delta}}
  \]
  - cf. to SGD discussion of previous slide

- **OPGD Algorithm**:
  \[
  w_{t+1} = P_S(w_t - \eta \nabla f_t(w_t))
  \]
  combines sub-gradients and projection onto \(S\)

- **Regret bound for OPGD**: when \(\|u\| \leq D\) \(\|\nabla f_t(u_t)\| \leq G\) and using \(\eta = \frac{D}{\sqrt{T}}\) we have \(R_T \leq DG\sqrt{T}\)
- **GD** (also denoted OGD) is the special case where \(S\) is unconstrained (e.g. \(R^d\)).
- The need to know \(T\) in advance (to set the learning rate) can be avoided by using a “doubling trick” and working in blocks of increasing \(T\) size for the same asymptotic bounds.

- **Follow The Leader (FTL)**:
  \[
  u_t = \arg\min_{u \in S} \sum_{i=1}^{t-1} f_i(u)
  \]
  - cf. ERM
  - FTL performs well in some cases (quadratic loss) but fails badly in others (e.g. linear loss)

- **Follow The Regularized Leader (FoReL)**:
  \[
  u_t = \arg\min_{u \in S} \sum_{i=1}^{t-1} R(u) + f_i(u)
  \]
  - cf. RLM

- **Regret bound for FTL**:
  \[
  R_T \leq \sum_{t=1}^{T} f_t(u_t) - f_t(u_{t+1})
  \]

- **Regret bound for FoReL**:
  \[
  R_T \leq R(u) + \sum_{t=1}^{T} f_t(u) - f_t(u_{t+1})
  \]

- **Regret bound for FoReL**: when \(f_t(\cdot)\) is \(L\)-Lipschitz and \(R(\cdot)\) is \(\sigma\)-strong convex \(R_T \leq R(u) + TL^2/\sigma\)
- By choice of \(R = \frac{1}{2} \| u \|^2\) and \(\sigma = \frac{1}{2}\) and \(\eta = \ldots\) obtain similar bound to OPGD
- But FoReL requires solving optimization problem from scratch at every step
- In the following we improve on this
• Regret bound through linearization:
• Let $z_t \in \nabla f_t(w_t)$. By properties of subgradients $f_t(w_t) - f_t(u) \leq z_t^T(w_t - u)$
• Fact: For any algorithm $R_T = \sum_{t=1}^{T} f_t(u_t) - \sum_{t=1}^{T} f_t(u) \leq \sum_{t=1}^{T} z_t^T(w_t - u)$
• This holds regardless of how the $w_t$ are produced
• and holds specifically if they are based on $z_t$
• FoRel's regret bound holds without change for the linearization of the $f_t$ (i.e., when it observes the $z_t$ and not the $f_t$)

• An aside: When $S = R^d$ (the unconstrained case) FoReL on the linearization of $f_t$ is identical to OGD. By the fact above this gives yet another regret bound for OGD.

• Running FoRel on the linearization can be easier
  
  \[ w_t = \arg \min_{w \in S} R(w) + \sum_{i=1}^{t-1} w^T z_i \]
  
  \[ = \arg \min_{w \in S} R(w) + w^T z_{1:t-1} \]

• This yields the OMD algorithm

• Online Mirror Descent:
• using notation of convex duality per $R^*$
  
  \[ g(\theta) = \arg \max_{w \in S} \ w^T \theta - R(w) = \nabla R^*(\theta) \]

• OMD:
  \[ \text{Init } \theta_1 = 0 \]
  \[ \text{Repeat: } w_t = g(\theta_t); \theta_{t+1} = \theta_t - z_t; t = t + 1 \]
• This is exactly FoRel on the linearization
• Enjoys the same regret bounds under generalized conditions
• Yields several known algorithms (and their analysis) as special cases
• Interesting variant: OGDLP (OGD with lazy projections) accumulates gradients in $\theta$ and projects the final result. Contrast this with OPGD which projects the result at every step and starts the next update from that point.
• OMD has direct analysis in terms of convex duality.