AdaBoost

Administrative/Meta:

- Homework 1 graded.
- Some references. The classical presentation of AdaBoost is Freund and Schapire (1997). It appears boosting was first described by von Neumann *[future matus: add the citation?]*. The analysis here, which follows the pattern of the smoothness analyses from the steepest descent lectures, can be extracted from (Telgarsky 2012) .

AdaBoost setup

Let's now look at the AdaBoost setup in more detail.

- **Loss** $\ell = \exp$ (exponential loss). It is also common to use the logistic loss $\ln(1 + \exp(\cdot))$, but for simplicity we'll use the standard choice.
- **Examples** $((x_i, y_i))_{i=1}^n$ with $x_i \in \mathcal{X}$ and $y_i \in \{-1, +1\}$. The main thing to note is that $\mathcal{X}$ is just some opaque set, we are not assuming vector space structure, and can not form inner products $\langle w, x\rangle$.
- **Elementary hypotheses** $\mathcal{H} = (h_j)_{j=1}^d$, where $h_j : \mathcal{X} \to [-1, +1]$ for each j. Rather than interacting with examples in $\mathcal{X}$ directly, boosting algorithms embed them in a vector space via these functions $\mathcal{H}$. For example, a vector $v \in \mathbb{R}^d$ is now interpreted as a linear combination of elements of $\mathcal{H}$, and predictions on a new example $x \in \mathcal{X}$ are computed as

  $$x \mapsto \sum_j v_j h_j(x).$$

  The method is called boosting because even if each $h \in \mathcal{H}$ is individually not a very good predictor, perhaps they can be merged into a linear combination which achieves good error. This is just like the earlier setup of learning a weight vector $w \in \mathbb{R}^d$, but now the predictor is in the subspace (in function space) spanned by $\mathcal{H}$. Many treatments consider $\mathcal{H}$ to be an infinite set (e.g., all decision trees), but we'll keep it finite for simplicity.
- **Data matrix** $A \in \mathbb{R}^{n\times d}$ no longer has $y_i x_i$ as row i, but instead

  $$A := \begin{bmatrix} y_1 h_1(x_1) & \cdots & y_1 h_d(x_1) \\ & \vdots & \\ y_n h_1(x_n) & \cdots & y_n h_d(x_n) \end{bmatrix}.$$

- **Empirical Risk** $\mathcal{L}\circ(-A)$, where $\mathcal{L}(v) := \frac{1}{n}\sum_{i=1}^n \ell(v_i)$; namely the empirical risk minimization problem is

  $$\min_{w\in\mathbb{R}^d} (\mathcal{L}\circ(-A))(w) = \min_{w\in\mathbb{R}^d} \mathcal{L}(-Aw) = \min_{w\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \ell(-(Aw)_i) = \min_{w\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \ell\left\{-y_i \sum_{j=1}^d h_j(x_i) w_j\right\}.$$

  This notation complete hides the class $\mathcal{H}$; however, it is essential to keep $\mathcal{H}$ in mind when considering the power and implementation of the method.

The AdaBoost algorithm is then $\|\cdot\|_1$ steepest descent, expanded with the present notation as follows.

1. Set $w_0 := 0$; this is a convention which allows the invariant that t iterations of boosting give w_t with at most t nonzero entries. This is desirable since the predictor $x \mapsto \sum_{j=1}^d w_j h_j(x)$ becomes expensive to compute as w becomes more dense.
2. For $s = 1, \ldots, t$:
   1. Choose
   $$v_s := \operatorname{argmax}\left\{\left\langle \nabla(\mathcal{L} \circ (-A))(w), v\right\rangle : \|v\|_1 \leq 1\right\} = \operatorname{argmax}\left\{\left\langle v, A^\top \nabla\mathcal{L}(-Aw)\right\rangle : \|v\|_1 \leq 1\right\}.$$
   This can be simplified further by noting that the maximum may be taken, without loss of generality, to be a coordinate $j \in [d]$ and a sign r. Namely, $v_s := r \cdot \mathbf{e}_j$, where
   $$\begin{aligned} j &:= \operatorname{argmax}_j \left|\left(A^\top \nabla\mathcal{L}(-Aw)\right)_j\right| \\ &= \operatorname{argmax}_j \left|\sum_{i=1}^n y_i \ell'(-(Aw)_i) h_j(x_i)\right|, \\ r &:= \operatorname{sgn}\left(\left(A^\top \nabla\mathcal{L}(-Aw)\right)_j\right). \end{aligned}$$
   Note that this optimization for j is a re-weighted ERM problem: the vector $\nabla\mathcal{L}(-Aw)$ provides the re-weighting. This argmax searches over $\mathcal{H}$ for a good choice $h_j \in \mathcal{H}$; namely, each round of AdaBoost selects another good hypothesis from $\mathcal{H}$.
   2. Update $w_i := w_{i-1} - \eta_i v_i$. Previously, the step size η_i was set to $\|\nabla(\mathcal{L} \circ (-A))\|_\infty / \beta$; here the choice will be somewhat more complicated, and deferred to the statement of the convergence rate.

Remarks.

- Recall from last lecture that the "easy scenario" (**separability**) was that there exists $v \in \mathbb{R}^d$ with $Av > 0$, which means a linear combinations of $\mathcal{H}$ which classifies every example correctly. This scenario was troubling because despite the apparent easiness, it implied non-existence of minima and lack of strong convexity, thus could not be attacked with the earlier tools. (Of course, we could switch to constrained optimization, noting that the exponential loss decays quickly, but it seems reasonable to analyze the algorithm as presented.)
- To be precise, the preceding point gave a choice of A which led to an optimization problem with certain properties beyond the convex optimization tools presented so far. It is natural to wonder if there are other choices of A which do satisfy strong convexity. Once again looking at $\|\cdot\|_2$ strong convexity (which implies $\|\cdot\|_1$ strong convexity, albeit with degraded constants), recall
$$\nabla^2(\mathcal{L} \circ (-A))w = A^\top \begin{bmatrix} \ell''(-(Aw)_1) & & \\ & \ddots & \\ & & \ell''(-(Aw)_n) \end{bmatrix} A.$$
We need to find some way to prevent the inner matrix going to zero; let's focus on this problem. As before, the inner matrix went to zero if there exists $v \in \mathbb{R}^d$ such that $Av > 0$; suppose instead that not only does such a v fail to exist, but suppose instead that every $v \in \mathbb{R}^d$ has coordinates i, i' such that $(Av)_i > 0$ and $(Av)_{i'} < 0$; namely at least one correct and one incorrect example. Such a condition suffices to give a form of strong convexity (Telgarsky 2012).

Convergence under separability

This section will prove a convergence rate of AdaBoost under the separability assumption, and still using the earlier steepest descent tools. The rate will be identical to the classical rate Freund and Schapire (1997), and

the step size is identical, meaning the iterate sequences are the same.

Before proceeding, there is another way to write the separability assumption, which will be more convenient mathematically. Say the **weak learning assumption (WLA)** is satisfied for matrix A with constant $\gamma > 0$ if

$$\inf_{\phi \in \mathbb{R}^n_+ \setminus \{0\}} \frac{\|A^\top \phi\|_\infty}{\|\phi\|_1} \geq \gamma.$$

Remark. Perhaps WLA looks strange, but note the following properties.

- WLA holds iff A is separable (there exists $v \in \mathbb{R}^d$ with $Av > 0$); thus WLA captures the "easy" scenario which eluded us before. (This "iff" is essentially the statement of Gordan's Theorem.)
- Note how this quantity is immediately beneficial to a convergence analysis. Recall that steepest descent analyses started from the upper bound

  $$f(w_{i+1}) \leq f(w_i) - \frac{\|\nabla f(w_i)\|_*^2}{2\beta},$$

  and then lower bounded $\|\nabla f(w_i)\|_*$ with some quantity related to $f(w_i) - f(\bar{w})$, thus leading to a recurrence relation which provided a convergence rate. Such an inequality is nearly provided by WLA, namely:

  $$\|A^\top \nabla \mathcal{L}(-Aw)\|_\infty \geq \gamma \|\nabla \mathcal{L}(-Aw)\|_1.$$

- Going a step further, notice how γ is related to the re-weighted ERM problem solved at each step of boosting:

  $$\max_j \left| \sum_{i=1}^n y_i \ell'(-(Aw)_i) h_j(x_i) \right| = \max_j |(A^\top \nabla \mathcal{L}(-Aw))_j| = \left\| (A^\top \nabla \mathcal{L}(-Aw))_j \right\|_\infty \geq \gamma \|\nabla \mathcal{L}(-Aw)\|_1.$$

  Morevoer, if $\nabla \mathcal{L}(-Aw) \neq 0$ (which would imply the first order conditions are satisfied and the method has converged), then both sides can be divided by $\|\nabla \mathcal{L}(-Aw)\|_1$, and thus the reweighting is a probability distribution, and WLA implies this reweighted ERM problem can always be solved to accuracy $1/2 + \gamma/2$. Said another way, WLA is a lower bound on the quality of $\mathcal{H}$ for *any* reweighting of $((x_i, y_i))_{i=1}^n$.

The convergence theorem is as follows. The step size is funny and will be described after the statement.

Theorem. Suppose WLA holds, and $(w_i)_{i=0}^t$ are generated by AdaBoost with step size $\eta_i := \|A^\top \nabla \mathcal{L}(-Aw_{i-1})\|_\infty / \|\nabla \mathcal{L}(-Aw_{i-1})\|_1 \geq \gamma$. Then

$$\mathcal{L}(-Aw_t) \leq \exp\left(-t\gamma^2/2\right).$$

Remarks.

- The step size deserves a fair bit of discussion. Since WL implies the $\inf_w \mathcal{L}(-Aw) = 0$, and since $\ell' = \ell$ when $\ell = \exp$, then the objective function becomes arbitrarily flat as iterations proceed. This step size is larger than the regular steepest descent choice $\|\nabla f(w_{i-1})\|_*/\beta$, which shrinks over time and is inadequate for this severely flat regime. The aggressive step size here is a key to the rate.
- It is also useful to recall the rate for λ-strongly-convex, β-smooth optimization:

  $$f(w_t) - f(\bar{w}) \leq (f(w_0) - f(\bar{w})) \exp(-t\lambda/\beta).$$

  Namely, the AdaBoost analysis replaces condition number β/λ with $2/\gamma^2$.

Let's now try to prove the theorem, proceeding from an upper bound resulting from smoothness as usual. Or, rather, almost as usual: let's slip in one detail, namely let's suppose the function is β_i smooth in iteration i, and that the step size is $\eta_i := \|A^\top \mathcal{L}(-Aw_{i-1})\|_\infty / \beta_i$. (By "smooth in iteration i", we mean that that

$\mathcal{L} \circ (-A)$ is β_i smooth over the sublevel set $\{w \in \mathbb{R}^d : \mathcal{L}(-Aw) \leq \mathcal{L}(-(Aw_{i-1}))\}$; these are the only points we care about in this iteration.) This, combined with WLA, gives

$$\begin{aligned}\mathcal{L}(-Aw_i) &\leq \mathcal{L}(-Aw_{i-1}) - \frac{\|A^\top \nabla \mathcal{L}(-Aw)\|_\infty^2}{2\beta_i} \\ &\leq \mathcal{L}(-Aw_{i-1}) - \frac{\gamma^2 \|\nabla \mathcal{L}(-Aw)\|_1^2}{2\beta_i}\end{aligned}$$

It appears we are stuck, but momentarily we will prove the following (which will return us to our earlier step size).

Lemma. For any w, w' with $\max\{\mathcal{L}(-Aw), \mathcal{L}(-Aw')\} \leq \mathcal{L}(-Aw_{i-1})$,

$$\left\|\nabla(\mathcal{L} \circ (-A))(w) - \nabla(\mathcal{L} \circ (-A))(w')\right\|_\infty \leq \mathcal{L}(-Aw_{i-1})\|w - w'\|_1.$$

In particular, this lemma says we can set $\beta_i := \mathcal{L}(-Aw_{i-1})$ (which recovers the step size from the theorem statement). Plugging this in and noting $\|\nabla \mathcal{L}(-Aw_{i-1})\|_1 = \mathcal{L}(-Aw_{i-1})$,

$$\begin{aligned}\mathcal{L}(-Aw_i) &\leq \mathcal{L}(-Aw_{i-1}) - \frac{\gamma^2 \|\mathcal{L}(-Aw_{i-1})\|_1^2}{2\mathcal{L}(-Aw_{i-1})} \\ &\leq \mathcal{L}(-Aw_{i-1}) - \frac{\gamma^2 \mathcal{L}(-Aw_{i-1})}{2}\end{aligned}$$

All that remains is to prove the lemma.

Proof (of preceding lemma)**.** *[future matus: explain the steps]*

$$\begin{aligned}\left\|\nabla(\mathcal{L} \circ (-A))(w) - \nabla(\mathcal{L} \circ (-A))(w')\right\|_\infty &= \max\left\{\left\langle Av, \nabla \mathcal{L}(-Aw) - \nabla \mathcal{L}(-Aw')\right\rangle : \|v\|_1 \leq 1\right\} \\ &\leq \frac{1}{n}\sum_{i=1}^n \left|\ell'(-(Aw)_i) - \ell'(-(Aw')_i)\right| \\ &= \frac{1}{n}\sum_{i=1}^n \left|\int_0^1 \ell''(-(Aw')_i + r((-Aw)_i - (-Aw')_i))(Aw - Aw')_i dr\right| \\ &\leq \left(\int_0^1 \frac{1}{n}\sum_{i=1}^n \ell''(-(Aw')_i + r((-Aw)_i - (-Aw')_i))(Aw - Aw')_i dr\right)\|Aw - Aw'\|_\infty \\ &\leq \left(\int_0^1 \frac{1}{n}\sum_{i=1}^n \ell(-(Aw')_i + r((-Aw)_i - (-Aw')_i))(Aw - Aw')_i dr\right)\|w - w'\|_1 \\ &\leq \mathcal{L}(-Aw_i)\|w - w'\|_1.\end{aligned}$$

□.

"Consistency" of convex ERM

The end of lecture gave a taste of the results in the next lecture. In the next lecture we'll close this segment on convex optimization by seeing how convex optimization helps us with our original problem, namely that of classification. We'll show the following two facts.

- For any $\epsilon > 0$, there exists a set of labeled training points so that for every convex loss ℓ with a positive subgradient at zero $s \in \partial\ell(0)$, the convex empirical risk minimizer has classification error $1 - \epsilon$. (This proof was briefly sketched in lecture.
- On the other hand, if we allow our class of prediction functions to grow, the convex empirical risk minimizer becomes achieves classification error arbitrarily close to the optimal choice.

References

Freund, Yoav, and Robert E. Schapire. 1997. "A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting." *J. Comput. Syst. Sci.* 55 (1): 119–39.

Telgarsky, Matus. 2012. "A Primal-Dual Convergence Analysis of Boosting." *JMLR* 13 (3): 561–606.