Boosting with Tempered Exponential Measures

Richard Nock Ehsan Amid Manfred K. Warmuth

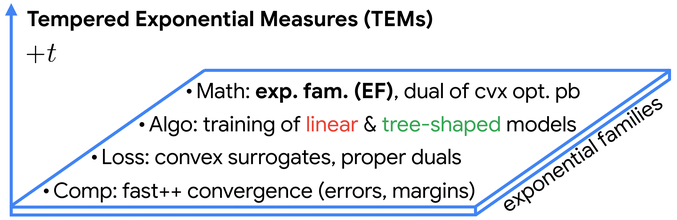
Google Research Google DeepMind

Summary

- Boosting fits parameters of an **exponential family (EF)**

$$\mathrm{pdf}(\boldsymbol{x}) \propto \exp(\boldsymbol{\mu}^\top \Upsilon(\boldsymbol{x}))$$

- We generalize it to a superset recently introduced, with ML benefits:



TEMs 101

Amid, Nock & Warmuth, 2023

- *t*-logarithm, *t*-exponential, *t*-arithmetic (here, $t \in [0,1]$)

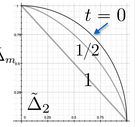
$$\log_t(z) \doteq \frac{1}{1-t} \cdot \left(z^{1-t} - 1\right), \quad \exp_t(z) \doteq [1 + (1-t)z]_+^{1/(1-t)}, \quad a \otimes_t b \doteq [a^{1-t} + b^{1-t} - 1]_+^{\frac{1}{1-t}}$$

$$[z]_+ \doteq \max\{0, z\} \qquad (t \to 1\text{: become "log", "exp" and "."})$$

- TEMs generalize / lift exponential families (EFs) in two directions

$$q_t(\boldsymbol{x}) \propto \exp_t(\boldsymbol{\mu}^\top \Upsilon(\boldsymbol{x})) \qquad \int q_t^{2-t} \mathrm{d}\xi = 1 \quad (q_t \in \tilde{\Delta}_m)$$

→ Normalized on *co-simplex* $\tilde{\Delta}_m$
→ *co-density* $p_t \doteq q_t^{2-t}$



The *t*-Boosting TEM

Boosting's EF: Kivinen & Warmuth, 1999

- We seek the following TEM (update)

$$\boldsymbol{q}' \doteq \arg \min_{\substack{\tilde{\boldsymbol{q}} \in \tilde{\Delta}_m \\ \tilde{\boldsymbol{q}}^\top \boldsymbol{u} = 0}} D_t(\tilde{\boldsymbol{q}} \| \boldsymbol{q}) \quad (1)$$

$$D_t(\boldsymbol{q} \| \boldsymbol{q}') \doteq \sum_{i \in [m]} q_i \cdot (\log_t q_i - \log_t q_i') - \log_{t-1} q_i + \log_{t-1} q_i'$$

tempered relative entropy (generalizes KL divergence)
edge vector $u_i \doteq y_i h(\boldsymbol{x}_i)$ (m examples, supervised learning +1/-1)

- Theorem: for $t \in \mathbb{R}_{\geqslant 0} \setminus \{2\}$,

→ solutions have the form

$$q_i' = \frac{q_i \otimes_t \exp_t(-\mu u_i)}{Z_t} \quad (2)$$

→ unknown μ satisfies

$$\mu = \arg\min Z_t(\mu') \quad \boldsymbol{q}'(\mu)^\top \boldsymbol{u} = 0 \quad (3)$$

→ $Z_t(\mu')$ is strictly convex*
→ finding solution to (1) via (2) easy using (3)

(*) +light assumption if $t = 0$

$$Z_t(\mu') = \left\| \boldsymbol{q} \otimes_t \exp_t(-\mu' \cdot \boldsymbol{u}) \right\|_{2-t}$$

Fitting the model part

- Models learned

$$\mathrm{H}_J(\boldsymbol{x}) \doteq \sum_{j \in [J]} \alpha_j h_j(\boldsymbol{x}) \qquad \mathrm{H}_J^{(\delta)}(\boldsymbol{x}) \doteq \sum_{\substack{(\delta) \\ (-\delta) \\ j \in [J]}} \alpha_j h_j(\boldsymbol{x}) \qquad h_j =$$

linear model **clipped** linear model decision tree

- Clipped summation

$$\sum_{\substack{(\delta) \\ (-\delta) \\ j \in [J]}} v_j \doteq \min \left\{ \delta, \max \left\{ -\delta, v_J + \sum_{\substack{(\delta) \\ (-\delta) \\ j \in [J-1]}} v_j \right\} \right\} \qquad v_1 \text{ if } J = 1 \quad (\in [-\delta, \delta])$$

(non commutative, "encoding-nice")

Example: $a = -1, b = 3, \delta = 2$
→ $v_1 = a, v_2 = b$
Clipped sum is 2 = -1 + 3
→ $v_1 = b, v_2 = a$
Clipped sum is **1** = **2** - 1

- Training the linear part: *t*-AdaBoost

Algorithm t-AdaBoost$(t, \mathcal{S}, J)$

Input: $t \in [0,1]$, training sample $\mathcal{S}$, #iterations J;
Output: classifiers $\mathrm{H}_J, \mathrm{H}_J^{(1/1-t)}$;
Step 1 : initialize tempered weights: $\boldsymbol{q}_1 = (1/m^{1/(2-t)}) \cdot \mathbf{1} \quad (\in \tilde{\Delta}_m)$;
Step 2 : for $j = 1, 2, ..., J$
  Step 2.1 : get weak classifier $h_j \leftarrow \mathrm{weak\_learner}(\boldsymbol{q}_j, \mathcal{S})$;
  Step 2.2 : choose weight update coefficient $\mu_j \in \mathbb{R}$;
  Step 2.3 : $\forall i \in [m]$, for $u_{ji} \doteq y_i h_j(\boldsymbol{x}_i)$, update tempered weights:

$$q_{(j+1)i} = \frac{q_{ji} \otimes_t \exp_t(-\mu_j u_{ji})}{Z_{tj}}, \quad \text{where } Z_{tj} = \left\| \boldsymbol{q}_j \otimes_t \exp_t(-\mu_j \cdot \boldsymbol{u}_j) \right\|_{2-t}.$$

  Step 2.4 : choose leveraging coefficient $\alpha_j \in \mathbb{R}$;

(remark that we allow $\alpha_j \neq \mu_j$)

Training the decision tree (DT) part = top-down splitting s with a twist-on-the-loss

- $(t = 1)$ AdaBoost → the most efficient splitting criterion for DT induction (Matushita's loss)
- We generalize to $t \neq 1$ and elicit a wide family of new losses for posterior estimation
- Process summarized (see paper for details):

$$Z_{tj} \xrightarrow{\text{models change}} \underline{L}^{(t)} \xrightarrow{\text{variational form}} \ell_1^{(t)}; \ell_{-1}^{(t)}$$

tempered exponential loss real-valued classification | pointwise Bayes risk class-probability estimation | partial losses

→ **tempered loss** $(t \in [-\infty, 2))$

$$\ell_1^{(t)}(u) \doteq \left(\frac{1-u}{M_{1-t}(u, 1-u)} \right)^{2-t} \qquad \ell_{-1}^{(t)}(u) \doteq \ell_1^{(t)}(1-u)$$

→ Power mean $M_q(a,b) \doteq \left(\frac{a^q + b^q}{2} \right)^{1/q}$
→ t=1: Matushita, t=0: Gini, t=-inf: error

Properties

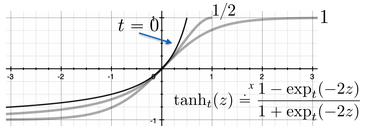
- (t-)margins

empirical margin risk

$$F_{t,\theta}(H, \mathcal{S}) \doteq \mathbb{E}_i[\![\nu_t((\boldsymbol{x}_i, y_i), H) \leqslant \theta]\!]$$

training sample

$$\nu_t((\boldsymbol{x}, y), H) \doteq \tanh_t(yH(\boldsymbol{x})/2)$$



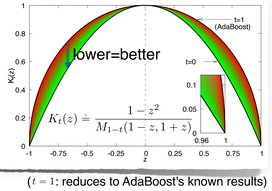
- Theorem (fast convergence for margins of t-AdaBoost, *simplified*), for $t \in [0,1]$

→ In t-AdaBoost, fix

$$\alpha_j \propto \mu_j \qquad \mu_j \propto -\log_t \left(\frac{1-\rho_j}{M_{1-t}(1-\rho_j, 1+\rho_j)} \right) \qquad \rho_j \propto \mathbb{E}_{\boldsymbol{q}_j}[y_i h_j(\boldsymbol{x}_i)] \quad (\in [-1,1])$$

→ If there is no "forgetting weights" (e.g. $\mathrm{H}_J, \mathrm{H}_J^{(1/1-t)}$ not good enough) then

$$F_{t,\theta}(H, \mathcal{S}) \leqslant \left(\frac{1+\theta}{1-\theta} \right)^{2-t} \cdot \prod_{j=1}^{J} K_t(\rho_j) \quad \text{for any } \theta \in (-1,1)$$



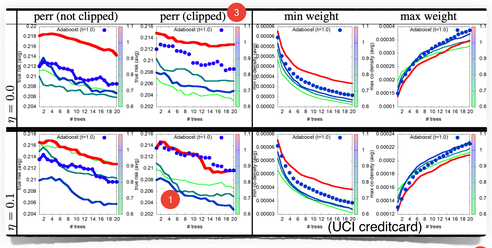
($t = 1$: reduces to AdaBoost's known results)

- Properties of tempered loss (DT induction), summarized

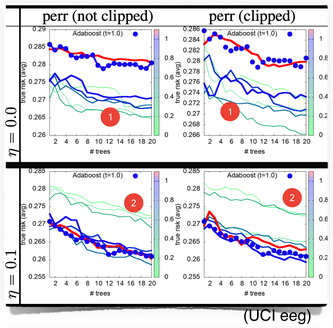
(Bayes rule optimal)
→ For any $t \in (-\infty, 2)$, the tempered loss is symmetric, differentiable and **strictly proper**
→ For $t = 2$, "just" symmetric and proper
→ Spans the full spectrum of know boosting rates for $t \in [-\infty, 1]$, near-optimal for $t = 1$... what about $t \in (1,2)$?

Experiments

- 10-folds stratified CV, different ts, t not in [0,1] and symmetric label noise $\eta \in \{0, 0.1\}$

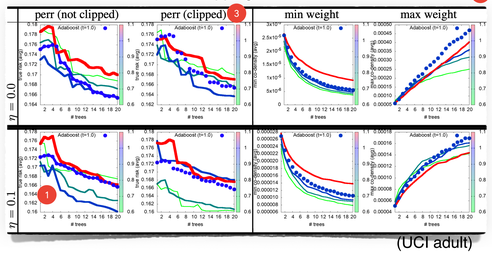


(UCI creditcard)



(UCI eeg)

→ $t \leqslant 1$ offers range of improvement compared to just AdaBoost (1)
→ +noise can completely reverse the picture (2)
→ $t > 1$ can be beneficial, clipping does work (3)



(UCI adult)

Conclusion:
- extension to more sophisticated boosting
- some properties *seem to* depend on t:
  - consistency of [early ! **no**] stopping
  - noise handling as a function of t
- tuning of t at training time
- code available at:
  http://users.cecs.anu.edu.au/~rnock/