Motivation: refreshing GLMs in production at reduced costs

- Production models at Criteo (GLMs) are refreshed around 4 times a day
- A single GLM learning takes 5 - 6 hours
- The training data from consecutive learnings (21 days, 3bln records, 3TB) overlap

Reducing learning time

How could we reuse the information from learning \( t \) in learning \( t+1 \) ?

Our approach a.k.a "model update" (MU)

As GLMs are smooth and convex, MU can be viewed as sequential
- Bayesian updates with a Laplace approximation of the weight posterior [1]
- batch learnings with a Taylor expansion of the loss (optimization* view)

Practically we propose to:
- carry over the Hessian \( H(\theta^{(m)}) \) with the optimal \( \theta^{(m)} \) at \( t \) for learning \( t+1 \)
- to limit the training data to non-overlapping sets at each iteration:
  - the 1st learning reads 21 days ("seed model")
  - the following learnings take varying training windows from 3h to 3 days

Mathematical derivation of MU (optimization view)

We assume that:
\[
\text{Loss}(D_{t+1}; \theta) = \text{Loss}(I_{t+1}; \theta) + \text{Loss}(D_t; \theta)
\]

Around \( \theta^{(m)} \) at learning \( t \) we form a 2nd order approximation of \( \text{Loss}(D_t; \theta) \) as
\[
\text{Loss}(D_t; \theta) = C + \frac{1}{2}(\theta - \theta^{(m)})^\top H(\theta^{(m)}) (\theta - \theta^{(m)})
\]

The final optimization problem for \( t+1 \) (solved through L-BFGS) is:
\[
\arg\min_{\theta^{(m)}} \text{Loss}(I_{t+1}; \theta) + \frac{1}{2}(\theta - \theta^{(m)})^\top H(\theta^{(m)}) (\theta - \theta^{(m)})
\]

The Hessian at \( t+1 \) is updated from the one at \( t \)

We measured the effect on the metrics over consecutive iterations for the Criteo bidding, recommendation, look and feel (L&F) products

Online behavior of MU on a Click prediction model

The model that was AB tested is a logistic loss predicting pClick given a display. This model, trained separately over 3 platforms (US, AS, EU), is used to predict:
- CTR (1)
- sales volume given a display (combination with other models) (2)
- sales amount given a display (combination with other models) (3)

Learning time and refresh frequency over all platforms
- Refresh: \( 1.6 \times \) more learnings for the MU configuration per day
- Cluster consumption: \( 9 \times \) speedup during learning for the MU configuration

Offline applications of MU for GLMs

We ran MU on logistic and geometric losses used at Criteo in an offline setting.

We measured the effect on the metrics over consecutive iterations for the Criteo

Online: Careful monitoring of drift is necessary
- Big win on cluster consumption and refresh frequency
- The diagonal approximation holds well in the first weeks but degrades over time
- Drift on CTR and long term metric per platform will be further investigated

Maintaining MU online would require scheduling automatic resets of the models.

Offline: MU can be applied to all GLM losses used at Criteo

The gains of MU offline are the biggest in the short term.
- Average speedup of \( \times 5 \)
- The effect of drift is negligible for a typical test configuration

Bibliography