Supplementary Material for: Learning Structured Models with the AUC Loss and Its Generalizations

Nir Rosenfeld
Hebrew University
Jerusalem, Israel

Ofer Meshi
TTI Chicago

Danny Tarlow
Microsoft Research
Cambridge, UK

Amir Globerson
Hebrew University
Jerusalem, Israel

1 Proof of Proposition 3.1

We have shown that with our enhanced representation, the ranking problem for given weights w reduces to the one in Joachims (2005) in the case of a fully-factored model. Here we show a similar result for the learning problem. Recall that our learning objective is defined as:

$$\min_w \frac{\lambda}{2}\|w\|^2 + \frac{1}{M}\sum_m \max_{z\in\mathcal{Z}} \left[w^\top \varphi(x^m, z) + \Delta_{AUC}(z, y^m)\right] - \max_{\substack{\hat{z}\in\mathcal{Z}\\ \hat{z}\sim y^m}} w^\top \varphi(x^m, \hat{z}) \quad (1)$$

We would like to consider this objective in the case of a fully-factored model, where the single element scores are simply: $w^\top x_i \equiv a_i$, and therefore: $w^\top\varphi(x,z) = \sum_i a_i \sum_k z_{ki}$.

Our goal is to show that the loss in this case is a simpler function of w. For a specific training example, we denote:

$$z^* = \operatorname*{argmax}_{z\in\mathcal{Z}} \left[w^\top\varphi(x^m, z) + \Delta_{AUC}(z, y^m)\right] \quad (2)$$

$$\hat{z}^* = \operatorname*{argmax}_{\hat{z}\in\mathcal{Z}, \hat{z}\sim y^m} w^\top\varphi(x^m, \hat{z}) \quad (3)$$

Due to the decomposition of the score and the AUC loss, z^* is obtained by sorting a vector which consists of elements $a_i - c$ for $i \in pos$ and a_j for $j \in neg$, with $c = 1/(|pos| \cdot |neg|)$. On the other hand, $\hat{z}^*$ is obtained by sorting the elements a_i for all $i \in pos$, then sorting the elements a_j for $j \in neg$, and then concatenating the results with pos before neg.

W.l.g. assume that the elements are indexed according to the ranking $\hat{z}^*$. We next define indicator variables for all pairs $i \in pos, j \in neg$:

$$y^*_{ij} = \begin{cases} 1 & a_i - c < a_j \\ 0 & \text{otherwise} \end{cases}$$

By construction, we know that i comes before j in $\hat{z}^*$. The variable y^*_{ij} represents whether i and j swapped positions (j is ranked before i) in z^*. We can now express the loss in this pairwise representation:

$$\begin{aligned} D(w) &= \sum_{i\in pos}\sum_{j\in neg} y^*_{ij}(w^\top(x_j - x_i) + c) - c\sum_{i\in pos}(n - i + 1) \\ &= \max_y \sum_{i\in pos}\sum_{j\in neg} y_{ij}(w^\top(x_j - x_i) + c) + const \end{aligned}$$

which is the same as Joachims (2005), up to additive constants (see proof of Lemma 2 therein).

2 LP Relaxation

In this section we show how to perform approximate inference in our model using LP relaxation. Recall that the problem we are interested in solving is:

$$\max_{z\in\mathcal{Z}} \sum_{ki}\theta_i(z_{ki}) + \sum_{kf}\theta_f(z_{kf}) \quad (4)$$

Equivalently, we can decompose the constraints $\mathcal{Z}$ into row and column constraints:

$$\max_{z\in\{0,1\}^{n\times n}} \sum_{ki}\theta_i(z_{ki}) + \sum_{kf}\theta_f(z_{kf}) + \sum_k R_k(z_{k\cdot}) + \sum_{k=1}^{n-1}\sum_i C(z_{ki}, z_{k+1,i})$$

where $R_k(z_{k\cdot}) = 0$ if $\sum_i z_{ki} = k$ and $-\infty$ otherwise, and $C(z_{ki}, z_{k+1,i}) = 0$ if $z_{ki} \le z_{k+1,i}$ and $-\infty$ otherwise. The natural LP relaxation is obtained by introducing marginal variables and relaxing the integrality constraints:

$$\begin{aligned} \max_{\mu\in\hat{\mathcal{L}}(G)} & \sum_{k,i}\sum_{z_{ki}} \mu_{ki}(z_{ki})\theta_i(z_{ki}) \\ &+ \sum_{k,f}\sum_{z_{kf}} \mu_{kf}(z_{kf})\theta_f(z_{kf}) \\ &+ \sum_k \sum_{z_{k\cdot}} \mu^R_k(z_{k\cdot})R_k(z_{k\cdot}) \quad (5) \\ &+ \sum_{k=1}^{n-1}\sum_i \sum_{z_{ki}, z_{k+1,i}} \mu^C_{ki}(z_{ki}, z_{k+1,i})C(z_{ki}, z_{k+1,i}) \end{aligned}$$

where $\hat{\mathcal{L}}(G)$ is the appropriate local polytope. One difficulty with the above formulation is that it involves an exponential number of variables μ^R corresponding to the row constraints. Instead, we propose to solve the more concise LP relaxation:

$$\begin{aligned} \max_{\mu\in\mathcal{L}(G)} & \sum_{k,i}\sum_{z_{ki}} \mu_{ki}(z_{ki})\theta_i(z_{ki}) + \\ & \sum_{k,f}\sum_{z_{kf}} \mu_{kf}(z_{kf})\theta_f(z_{kf}) \\ \text{s.t.} & \sum_i \mu_{ki}(1) = k; \quad \mu_{k+1,i}(1) \geq \mu_{ki}(1) \end{aligned} \tag{6}$$

We next show that this compact LP is in fact equivalent to the more expensive one.

To prove this, we show that the constraint matrix in Eq. (6) is totally unimodular (TUM), which guarantees that all vertices of the corresponding polytope are integral. Since it is clear that any (integral) solution of the full representation is also a solution in the compact representation, proving unimodularity will guarantee that they are in fact equivalent.

Consider the constraint set on μ:

$$\sum_i \mu_{ki}(1) = k, \qquad \mu_{k+1,i}(1) \geq \mu_{ki}(1) \tag{7}$$

These linear constraints can be expressed in matrix form by $A\mu \leq b$ for the appropriate A, b. By rearranging rows and columns (to which TUM is invariant), A can be written as:

$$A = \begin{pmatrix} I & I & \cdots & I \\ -I & -I & \cdots & -I \\ G & 0 & \cdots & 0 \\ 0 & G & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G \end{pmatrix}$$

$$G = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \end{pmatrix}$$

Where I is of size $n \times n$ and G of size $(n-1) \times n$. To show that A is TUM, we use the Ghouila-Houri sufficient condition (see Schrijver, 2003). The condition is as follows: A matrix A is TUM iff for every subset R of rows, there is an assignment $s : R \to \pm 1$ such that $\sigma_i = \sum_{r\in R} A_{ri}s(r) \in \{0, \pm 1\}$ for every column i.

Proof: Let $R = (R_I, R_{-I}, R_{G_1}, \ldots, R_{G_n})$ be a subset of rows of A (we use the subscripts to relate rows to their corresponding blocks in A). We start with $\sigma = 0$, and sequentially add rows and update σ. First, we set $s(u) = (1, 1, \ldots, 1)$ for every row u in R_I, R_{-I}, and compute the sum over these rows. Denote by v the vector attained by this sum. Since each column (constrained to these rows) contains zeros and either 1,-1 or both, all entries in v are in $\{0, \pm 1\}$.

Notice that since the remaining part in A is block-diagonal, the row sets R_{G_i} are independent with respect to σ. Hence, we focus on some arbitrary R_G, and show a choice of s_G under which the entries in σ corresponding to G's columns are in $\{0, \pm 1\}$.

Let i be the index of the first row in R_G:

- If $v_i \neq 0$, assume w.l.g that $v_i = 1$. We set $s_G(i) = -1$ to ensure that $\sigma_i = 0$. For all adjacent trailing rows $i < j \in R_G$, we also set $s_G(j) = -1$. Consider adding row j to σ. If $v_j = 0$, then after the addition $\sigma_j = 0$. Otherwise, $\sigma_j = -1$. When adding row $j+1$, $\sigma_j = 0$, and σ_{j+1} is either 0 or -1, depending on v_{j+1}. This invariant is true for all sequentially added rows.
- Otherwise, if $v_{i+1} \neq 0$, assume w.l.g. that $v_{i+1} = 1$. Similarly, we set $s_G(i) = +1$ to ensure that $\sigma_i = 0$. Again, adding rows j sequentially results in $\sigma_j \in \{0, 1\}$.
- Otherwise, we are free to choose any sign for i.

Repeating this process for all of other G-block rows results in $\sigma \in \{0, \pm 1\}$ as desired.

3 Experiments with AUC Generalizations

We suggested three generalizations to the AUC - Binned AUC, AUC@k, and Unnested AUC. Here we present experiments which test their relation to the standard AUC. In each task we compare models trained with the standard AUC loss to models trained with a generalized AUC loss. We then compare how each model performs on both measures. Results are presented in Table 3.

The binned AUC measure is suitable for tasks where items can be grouped into bins, where inter-bin ordering is of little importance. Document retrieval is a natural candidate since documents are often presented to end users in sequential pages. Hence, we use the binned AUC measure on the first fold of the OHSUMED dataset. As expected, binned AUC scores are higher than standard AUC scores for both models (some of the swapped pairs are not counted). Surprisingly, binned AUC for the binned model is lower

Table 1: Performance on standard AUC and its generalizations. Columns indicate which loss the model was trained with, and rows indicate which measure was used for evaluation. TOP: Binned AUC for factored models on the OHSUMED dataset. CENTER: AUC@k for factored models on the original NIPS dataset. BOTTOM: Unnested AUC for pairwise models on the Yeast multilabel dataset.

OHSUMED

Test \ Train	AUC	Binned
AUC	63.1	62.4
Binned	68.9	68.6

NIPS

Test \ Train	AUC	AUC@k
AUC	70.4	58.2
AUC@k	77.1	77.4

Yeast

Test \ Train	AUC	Unnested
AUC	82.44	82.51
Unnested	82.35	82.59

than for the un-binned model. This may be due to the non-convexity of the binned objective.

The AUC@k measure is suitable for tasks where only the top k ranked items are of interest. In online social networks, users are often presented with a short list of people whom they are likely to add as friends. Hence, we test the AUC@k measure on the task of link prediction in the original NIPS dataset[1] (years 1987-2003). Optimizing over the binned loss results in better binned AUC but worse standard AUC.

Unnested AUC is suitable for tasks where increasing the number of 'on' variables may result in changing previous choices. We test the unnested AUC on the Yeast multi-label dataset[2], since many samples have several labels. Our experiments show that in this setting, allowing unnested outputs increased performance only slightly. We deduce that while outputs where not constrained to be nested, most of them in fact were.

References

T. Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384. ACM, 2005.

A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Verlag, 2003.

[1] `http://ai.stanford.edu/~gal/data.html`

[2] Taken from the Mulan collection: `http://mulan.sourceforge.net/datasets.html`