Online Active Learning for Prec@K

Aditya Modi, Ayush Sekhari, Chirag Gupta

December 4, 2017

1 Introduction

We aimed to take up the problem of reducing label complexity in online algorithms for structured prediction problem. As various methods in structured prediction work through devising a proxy for the loss to be minimized, we take up an instance of this bigger picture. A structured estimation problem can be seen as finding a corresponding $y \in \mathcal{Y}$, from a set of complex labels for each input instance $x \in \mathcal{X}$. This is done by finding a function $s(x, y)$ with

$$y^* = \texttt{argmax}_{y \in \mathcal{Y}} f(x, z)$$

We consider the problem of ranking where each instance is a set of points x_i and the mapping y corresponds to a permutation of these points. We have a variety of measures for assessing the performance of the mapping returned by a learner like maximum average precision (MAP), precision@k (prec@k), Normalized discounted cumulative gain (NDCG), mean reciprocal rank, winner takes all (WTA) etc. Specifically, we look at the problem of bipartite ranking and choose a performance measure known as prec@k for the problem. Apart from being a special case of the structured prediction task, we get few motivating properties for activizing the methods as well. In bipartite ranking, the input set is a collection of positive (relevant) and negative (irrelevant) points. With prec@k as the metric, the goal is to rank any subset of size k at the top. We see the problem in itself needs to identify a very small subset of positive points. Instead of labelling each instance in the input, we can intuitively benefit by concentrating on this small number k. In the next section, we describe some related work and in further section mention the direction of our work.

2 Related work

In the bipartite problem, let $\mathbf{X}_+$ and $\mathbf{X}_-$ denote the positive and negative instances. Popular algorithms learn a score $s(x) \in \mathbf{R}$ for each $x \in \mathcal{X}$ and the output is given by sorting the points according to this score. With the returned permutation $\sigma(s)$, the loss for prec@k is defined as

$$prec@k(s, x_1, x_2, \ldots, x_n) = \Delta(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^{k} (1 - y_{\sigma_s(i)})$$

Notice that this is a highly non-convex optimization problem and therefore, direct minimization of this metric is not possible. In a recent paper, Kar et al. [KNJ15] give three surrogates for prec@k which upper bound the loss and are consistent in specific margin conditions. Margin based models have been very successful in classification problems and provide a framework for active learning as well (refer [BBZ07]). The surrogate loss as given by the authors intuitively penalize the algorithm for those negative vectors which are ranked above the top scored positives. The tightest surrogate, $l^{\texttt{ramp}}_{\texttt{prec@k}}$ ramp surrogate as they call it, is as follows:

$$\texttt{max}_{\|\hat{\mathbf{y}}\|_1=k}\{\Delta(\mathbf{y},\hat{\mathbf{y}}) + \sum_{i=1}^{n}\hat{\mathbf{y}}_i s_i\} - \texttt{max}_{\|\tilde{\mathbf{y}}\|_1=k, K(\mathbf{y},\tilde{\mathbf{y}})=k}\sum_{i=1}^{n}\tilde{\mathbf{y}}_i s_i \tag{1}$$

where $\hat{\mathbf{y}}$ is the returned label for the instance and s_i denotes the score for $\mathbf{X}_i$. A corresponding margin criteria for it is that some k positive points considerably outrank all negatives, i.e., for some scoring function s and set S_+, $\texttt{min}_{i\in S_+, |S_+|=k} s_i - \texttt{max}_{j:\mathbf{y}_j=0} s_j \geq \gamma$. This gives the most relaxed condition for this problem. To do away with the non-convex nature of the ramp surrogate, they propose two more surrogates: max surrogate and avg-surrogate. The max-surrogate is given as:

$$\texttt{max}_{\|\hat{\mathbf{y}}\|_1=k}\{\Delta(\mathbf{y},\hat{\mathbf{y}}) + \sum_{i=1}^{n}(\hat{\mathbf{y}}_i - \mathbf{y}_i)s_i + \texttt{max}_{\|\tilde{\mathbf{y}}\|_1=n_+-k, \tilde{\mathbf{y}}\preceq(1-\hat{\mathbf{y}})\cdot\mathbf{y}}\sum_{i=1}^{n}\tilde{\mathbf{y}}_i s_i\} \tag{2}$$

The margin condition for this is the regular binary classification margin condition. Kar et al. propose perceptron based learning methods for the convex surrogates. We consider the algorithm for max-surrogate as it offers us a readily available advantage for sampling only a few false negatives along with top-k labels. The algorithm is shown in 1. Similar to the mistake bound with hinge loss in classification, we get the following mistake bound:

Theorem 1. ***Mistake bound for Perceptron@K-max*** *Let* $\|x_t^i\| \leq R \quad \forall t, i$. *With* $\Delta = \sum_{t=1}^{T}\Delta_t$ *be the cumulative observed mistake according to algorithm 1 for* T *rounds, we have*

$$\Delta \leq \texttt{min}_{\mathbf{w}}(\|\mathbf{w}\|R\sqrt{4k} + \sqrt{\hat{\mathcal{L}}_T^{max}(\mathbf{w})}) \tag{3}$$

where $\hat{\mathcal{L}}_T^{max}(\mathbf{w})$ *is the cumulative loss with max-surrogate.*

We note that the formulation is very similar to the regular perceptron learning rule. We therefore refer to two primary works in active learning for perceptron models: randomized selective sampling [CbGZ05] and active perceptron [DKM09]. In [CbGZ05], Cesa-Bianchi et al. propose randomizing the query for labels for each point by taking the score $s = \langle \mathbf{w}, \mathbf{x}\rangle$ and defining a Bernoulli random variable with $p = \frac{b}{b+s}$. If the label is not queried, no update is made. They show that, in expectation, this method achieves the same mistake bound as the usual perceptron rule. Seeing the proximity of our method to perceptron, we devise a few sampling schemes which are described in the next section. However, this does not give us any relation between the number of mistakes made and the number of labels queried. We would want to combine these two quantities to obtain a meaningful upper bound. As such, we also look at the active perceptron model. In the active perceptron model, Dasgupta et al. propose

a threshold based query strategy based on the score returned by the model. After scaling the update rule with the factor $2\langle \mathbf{w}, \mathbf{x} \rangle$, they show an improvement to $\tilde{\mathcal{O}}(d\texttt{log}\frac{1}{\epsilon})$ label complexity for maximum generalization error ϵ. We aim to reduce the query complexity further by using such a strategy based on the margin criteria. This would enable us to reduce the dependency of the query complexity on k as well. In the next section, we describe the proposed selective sampling strategy and show some empirical results in the following section.

3 Active Perceptron for Prec@K

We pick a simple algorithm proposed in [KNJ15]. We then propose a sampling scheme to ascertain whether or not to query for the label of a data point or not. This sampling scheme is a function of the margin that the classifier enjoys to separate it from the top K data points. Algorithm 1 is the original algorithm, and algorithm 2 is the active one.

Data: N data points, value of K
Randomly divide data into $log(N)$ batches of size $N/log(N)$;
$\mathbf{w} \leftarrow 0$;
while *stream active* **do**
  read current batch $\mathbf{B} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_b]$;
  calculate scores $\mathbf{s} = \mathbf{w}^T\mathbf{B}$;
  return top K values of $\mathbf{s}$ as prediction;
  observe *complete* labels $\mathbf{y} \in \{0,1\}^b$, incur prec@k loss, say Δ;
  if $\Delta \neq 0$ **then**
    for *every false positive* $\mathbf{x}$ *in the top* K **do**
      $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{x}$
    end
    for *the top* Δ *false negatives not in the top* K *according to* $\mathbf{s}$ **do**
      $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{x}$;
    end
  end
end

Algorithm 1: Original algorithm

Data: N data points, value of K
Randomly divide data into $log(N)$ batches of size $N/log(N)$;
$\mathbf{w} \leftarrow 0$;
while *stream active* **do**
  read current batch $\mathbf{B} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_b]$;
  calculate scores $\mathbf{s} = \mathbf{w}^T\mathbf{B}$;
  return top K values of $\mathbf{s}$ as prediction;
  query *all* labels $\mathbf{y} \in \{0,1\}^b$, incur prec@k loss, say Δ;
  if $\Delta \neq 0$ **then**
    for *every false positive* $\mathbf{x}$ *in the top* K **do**
      $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{x}$
    end
    for *every* $\mathbf{x}$ *not in the top* K *according to* $\mathbf{s}$ **do**
      query $\mathbf{x}$ with probability $\propto exp(s[\text{point ranked } k] - s[\mathbf{x}])$;
      if *query was made* **then**
        $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{x}$;
    end
  end
end

Algorithm 2: Active algorithm

3.1 Sampling schemes

While sampling, we essentially want points closer to the margin to be queried with higher probability since this is the region where we expect to find false negatives. Why not just query for a fixed number of points (say Δ) after the top K? This is because of two reasons -

- Sampling makes the algorithm more robust to adversarial or noisy data.
- If the margin is sufficiently high, it is possible that we don't sample at all, since every sample is drawn individually.

In general, we expect to obtain better regret and query complexity guarantees with sampling. We present results on four sampling schemes, out of which two of them are baselines schemes.

1. **Sequential sampling from the top.** We keep sampling from the top until we obtain Δ many false negatives. This is not really a sampling scheme, in fact it is completely equivalent to algorithm 1. This scheme allows us to compare against the minimum number of queries required for the original algorithm.
2. **Exponentially decaying sampling.** This is the sampling scheme proposed in 2. Points closer to the margin are sampled with exponentially higher probability.

$$P(\text{querying a point}) \propto exp(\text{score of that point} - \text{score of point ranked k})$$

3. **Inverse sampling.** This is an alternative sampling scheme to scheme 2.

$$P(\text{querying a point}) \propto 1/(\text{score of point ranked k} - \text{score of that point})$$

4. **Uniform sampling.** This is a baseline scheme. Every point not in the top K is sampled uniformly. This is a baseline scheme; we must definitely beat this if our notion of margin is useful.

4 Preliminary Experiments

We performed experiments on the COD-RNA [UKM06] data-set which has the reputation of being an easy data-set for ranking and classification tasks.

4.1 Data-Set Description [COD-RNA]

COD-RNA is a numerical data-set with 488565 data points (162855 positive and 325710 negative). Each feature vector is 8 dimensional. We used mean-normalized data for our experiments.

4.2 Experiments

We compared the Prec@K(for each batch) and cumulative number of queries required by our algorithms for bipartite ranking over the COD-RNA dataset.(For Algorithm 1, this quantity is the number of points to be queried to get top Δ number of false negatives to update the perceptron algorithm). Figure 1 and Figure 3 correspond to Prec@50 with and without kernelization respectively. Figure 2 and Figure 4 correspond to Prec@300 with and without kernelization respectively.
The four algorithms plotted in the presented results are described as follows (refer to section 3.1 for a description of the sampling schemes):

1. **Algorithm - 1**: Non-active algorithm of [KNJ15] (sampling Scheme 1).
2. **Algorithm - 2**: Proposed online active algorithm using sampling scheme 2.
3. **Algorithm - 3**: Proposed online active algorithm using sampling scheme 3.
4. **Algorithm - 4**: Proposed online active algorithm using sampling scheme 4.

We found that our algorithms were working comparable to the state of the art non-active algorithm of [KNJ15] in much fewer number of active queries. We also observed that kernelization with RBF kernels significantly improved the performance of both algorithms. We also observed that using a probability function which decays with margin performs better than uniform active sampling and is stabler to data variations across mini-batches, which emphasizes use of margin for actively querying. We expect to be able to prove that our algorithm, when operated over mini-batches of size $\frac{n}{log(n)}$ in an actively online fashion would require $\mathcal{O}(k \cdot log(n))$ number of label queries on expectation / with high probability.

5 Future work

1. We mentioned that COD-RNA was a simple data-set to work with. To truly validate our algorithm, we intend to perform experiments with difficult data-sets such as COVTYPE. Another issue with COD-RNA is that the class imbalance problem is very benign here (the class ratio being 1:2). We are interested in observing the performance of our algorithm when the class ratio is more stringent.
2. The work as of now is devoid of formal guarantees in terms of query complexity and regret. This will be an important aspect of our work.
3. We believe that with better sampling schemes, we should be able to reduce query complexity from $\mathcal{O}(k) \cdot o(n)$ to $o(k) \cdot o(n)$.
4. The algorithm is currently stochastic in the sense that it assumes that the entire batch of size n is available, and then proceeds to divide it into mini-batches. What if data is being streamed in an online fashion. We will need algorithmic and theoretical solutions for the online setting as well.

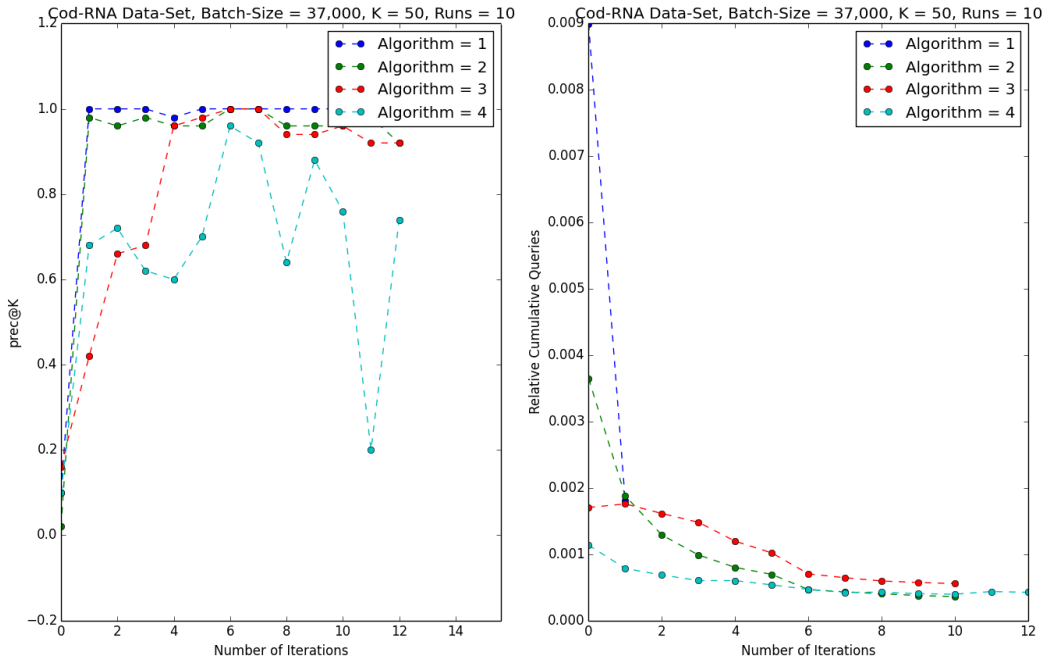

Figure 1: Prec@50 for COD-RNA [unkernelized algorithms]

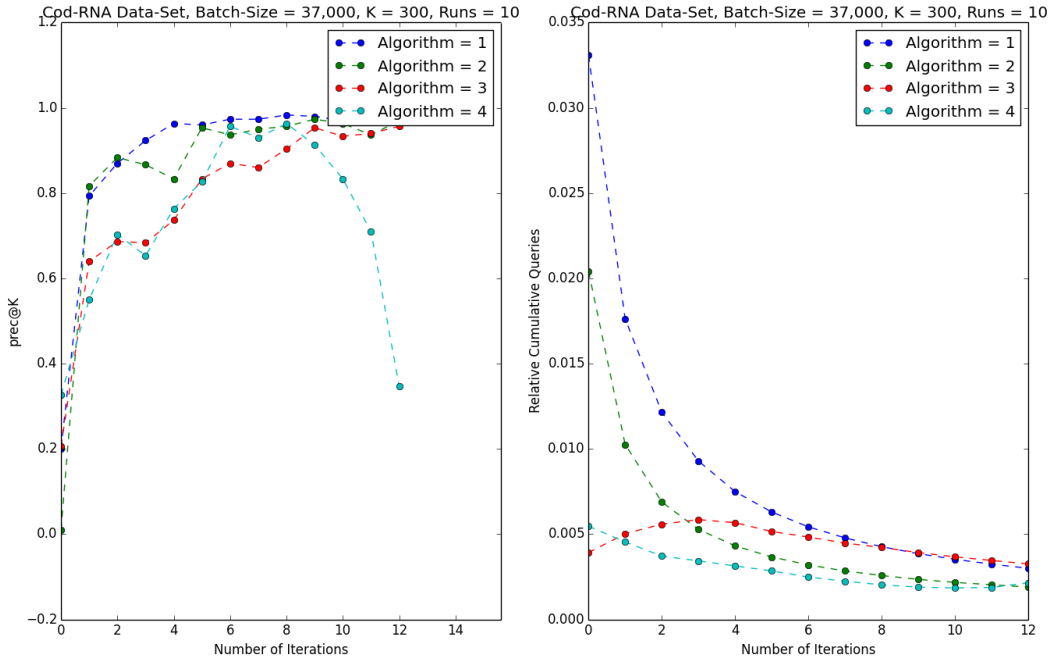

Figure 2: Prec@300 for COD-RNA [unkernelized algorithms]

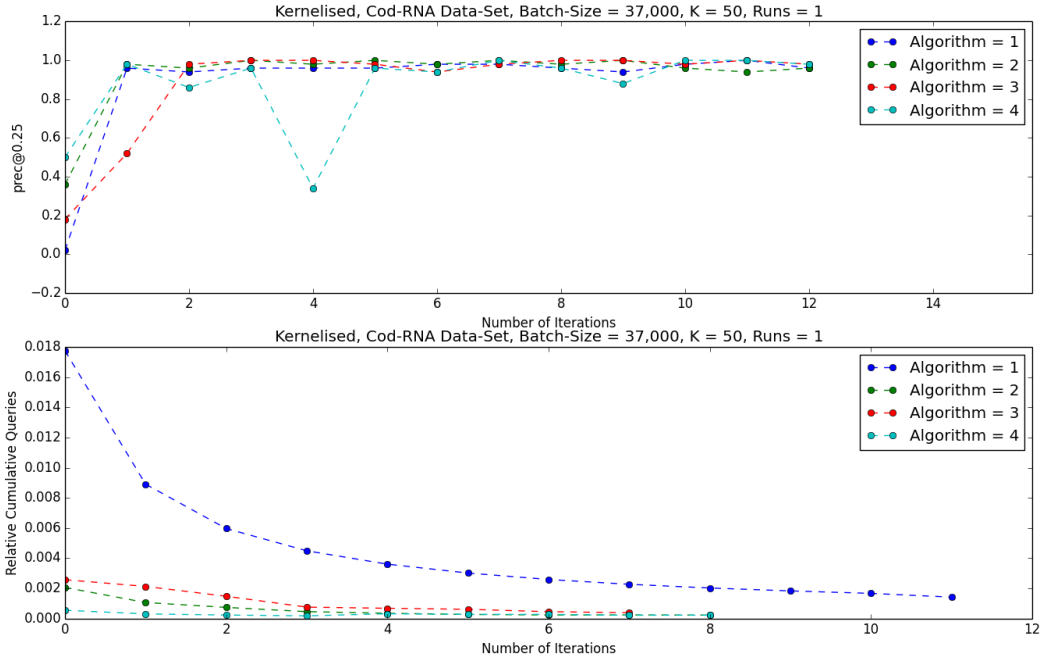

Figure 3: Prec@50 for COD-RNA [kernelized algorithms]

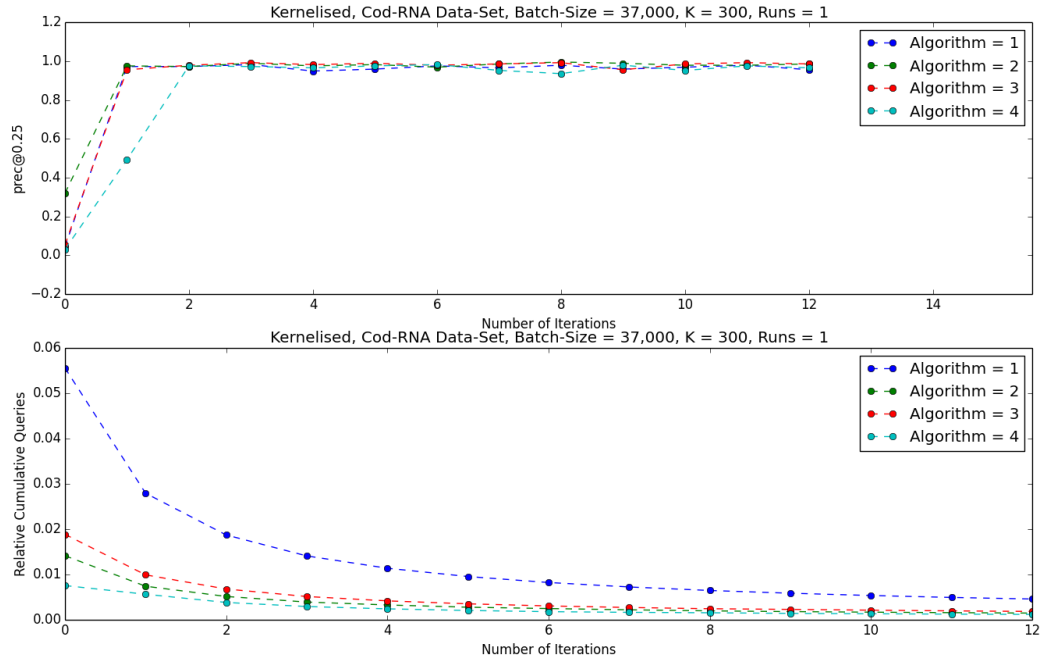

Figure 4: Prec@300 for COD-RNA [kernelized algorithms]

References

[BBZ07] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *Learning Theory*, pages 35–50. Springer, 2007.

[CbGZ05] Nicolò Cesa-bianchi, Claudio Gentile, and Luca Zaniboni. Worst-case analysis of selective sampling for linear-threshold algorithms. In *Advances in Neural Information Processing Systems*, pages 241–248, 2005.

[DKM09] Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. *The Journal of Machine Learning Research*, 10:281–299, 2009.

[KNJ15] Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Surrogate functions for maximizing precision at the top. *arXiv preprint arXiv:1505.06813*, 2015.

[UKM06] Andrew V Uzilov, Joshua M Keegan, and David H Mathews. Detection of noncoding rnas on the basis of predicted secondary structure formation free energy change. *BMC bioinformatics*, 7(1):1, 2006.