Scalable Learning of Non-Decomposable Objectives

Elad Eban
elade@google.com

Mariano Schain
marianos@google.com

Alan Mackey
mackeya@google.com

Ariel Gordon
gariel@google.com

Rif A. Saurous
rif@google.com

Gal Elidan
elidan@google.com

Google, Inc.

Abstract

Modern retrieval systems are often driven by an underlying machine learning model. The goal of such systems is to identify and possibly rank the few most relevant items for a given query or context. Thus, such systems are typically evaluated using a ranking-based performance metric such as the area under the precision-recall curve, the F_β score, precision at fixed recall, etc. Obviously, it is desirable to train such systems to optimize the metric of interest.

In practice, due to the scalability limitations of existing approaches for optimizing such objectives, large-scale retrieval systems are instead trained to maximize classification accuracy, in the hope that performance as measured via the true objective will also be favorable. In this work we present a unified framework that, using straightforward building block bounds, allows for highly scalable optimization of a wide range of ranking-based objectives. We demonstrate the advantage of our approach on several real-life retrieval problems that are significantly larger than those considered in the literature, while achieving substantial improvement in performance over the accuracy-objective baseline.

1 Introduction

Machine learning models underlie most modern automated retrieval systems. The quality of such systems is evaluated using ranking-based measures such as area under the ROC curve (**AUCROC**) or, as is more appropriate in the common scenario of few relevant items, measures such as area under the precision recall curve (**AUCPR**, also known as average precision), mean average precision (**MAP**), precision at a fixed recall rate (**P@R**), etc. In fraud detection, for example, we would like to constrain the fraction of customers that are falsely identified as fraudsters, while maximizing the recall of true ones.

What is common to all of the above objectives is that, unlike standard classification loss, they only partially or do not at all decompose over examples. This makes optimization more difficult and consequently machine learning retrieval systems are often not trained to optimize the objective of interest. Instead, they are typically trained simply to maximize classification accuracy in the hope that the retrieval performance will also be favorable. Unfortunately, this discrepancy can lead to inferior results, as is illustrated in Figure 1 (see also, for example, [7, 9, 27]). Our goal in this work is to develop a unified approach that is applicable to a wide range of rank-based objectives and that is scalable to the largest of datasets, i.e. that is as scalable as methods that optimize for classification accuracy.

Several recent works, starting with the seminal work of Joachims [11] have addressed the challenge of optimizing various rank-based objectives. All of these, however, still suffer from computational scalability issues, or are limited to a specific metric. Joachims [11], for example, offers a method for optimizing $\mathbf{F}_\beta$ and **P@R**. In general, the computation of each gradient step is quadratic in the number of training instances, and slow even in the best of cases. As another example,

Yue et al. [27] optimize for **MAP** but are hindered by the use of a costly cutting plane training algorithm. See section 2 for a more detailed list of related works and discussion of the scalability limitations.

In this work, we propose an alternative formulation that is based on simple direct bounds on per-example quantities indicating whether each example is a true positive or a false positive. These building block bounds allow us to construct global bounds on a wide range of ranking based, non-decomposable objectives, including all of those mentioned above. Importantly, the surrogate objectives we derive can be optimized using standard stochastic (mini-batch) gradient methods for saddle-point problems with favorable convergence rates [6]. This is a decisive advantage at the massive scale on which most modern automated retrieval systems must operate, where methods requiring full-batch optimization are intractable.

Following the development of the bounds for a range of measures, we demonstrate the effectiveness of our approach for optimizing **AUCPR** and other objectives on several real-life problems that are substantially larger than those considered in the literature. Empirically, we observe both improvement in performance compared to the use of standard loss functions (log-loss), and a favorable convergence rate indistinguishable from the vanilla SGD baseline rates.

Our contribution is thus threefold. First, we provide a unified approach that, using the same building
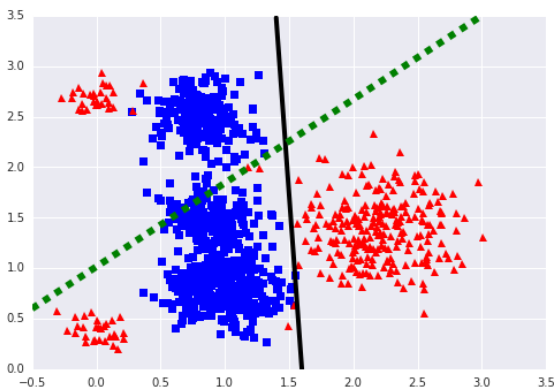

Figure 1: Illustration of the potential difference between classification accuracy and, for example, the max **P@R** = 0.95 objective. The red triangles and blue squares represent positive and negative examples respectively. The black solid line corresponds to the best linear classifier which is 90% accurate. If the threshold of this classifier is changed to achieve recall of 95%, the precision will be 50%. The dashed green line corresponds to a classifier that achieves a recall of 95% but with precision of about 65%.

blocks, allows for the optimization of a wide range of rank-based objectives that include **AUCROC**, **AUCPR**, **P@R**, **R@P**, and $\mathbf{F}_\beta$. Third, our unified framework also easily allows for novel objectives such as the area under the curve for a region of interest, i.e. when the precision or recall are in some desired range. Finally, and most importantly, our bounds give rise to an optimization approach for non-decomposable learning metrics that is highly scalable and that is applicable to truly large datasets.

2 Related Works

Several works in the last decade have focused on developing methods for optimizing rank-based objectives. Below we outline those most relevant to our work while highlighting the inherent scalability limitation which is our central motivation.

The seminal paper of Joachims [11] uses a bound on the possible number of contingency tables to optimize $\mathbf{F}_\beta$ and **P@R**, and a bound based on individual pairs of examples to optimize **AUCROC**; in both cases the system iteratively solves a polynomial-time optimization sub-problem to find a constraint to add to a global optimization, which generalizes structured SVMs [25]. The scalability of this approach is limited since the cost of computing a single gradient is generally quadratic (and always super-linear) in the number of training examples. Furthermore, even in the best case, Joachims' loss function and its gradient take at least linear time to compute, resulting in a slow gradient-descent algorithm. This is in contrast to our stochastic gradient approach.

Optimizing the **AUCPR** or the related mean-average-precision is, in principle, even more difficult since the objective does not decompose over pairs of examples. [13] and [4] tackle this objective directly, and [27] proposes a more efficient AP-SVM which relies on a hinge-loss relaxation of the **MAP** problem. While the work of [27] demonstrates the merit of optimizing **MAP** instead of accuracy for reasonably sized domains, scalability is still hindered by the use of a cutting plane training algorithm that requires a costly identification of the most violated constraint. To overcome this, [15] suggests several innovative heuristic improvements that achieve appealing running time gains, but do not inherently solve the underlying scalability problem. [23] generalizes the above approaches to the case of nonlinear deep network optimization; their approach "has the same complexity" as [27], and is thus also not scalable to very large problems.

Other approaches achieve scalability by considering a restricted class of models [14] or targeting only specific

objectives [5, 10, 16, 18, 20]. The work of [19] achieves both scalability and generality, but does not cover objectives which place a constraint on the model, such as recall at a fixed precision, precision at a fixed recall, or accuracy at a quantile. [2] optimize this latter objective in an elegant method that is theoretically scalable, since it can be distributed to many machines. However, it requires solving an optimization problem per instance and is thus not truly scalable in practice. Finally, [12] propose a general purpose approach for optimizing non-decomposable objectives. Their method is cast in the online setting, and the adaptation they suggest for stochastic gradient optimization with minibatches requires a buffer. For extremely multi-label problems such as one we consider below, maintaining such a buffer may not be possible.

3 Building Block Bounds

In this section we briefly describe the simple building block bounds of the true positive and false positive quantities. These statistics will form the basis for the objectives of interest throughout our work.

We start by defining the basic entities involved in rank-based metrics. We use X to denote the explanatory features, Y to denote the target label, Y^+ to denote the positive examples, and Y^- to denote the negatives.

Definition 3.1. A classification rule f_b is characterized by a score function $f : X \to \mathbb{R}$, and a threshold $b \in \mathbb{R}$, indicating that classification is done according to $f(x) \geq b$.

Note that we intentionally separate the parameters of the models embedded in f (which could be a linear model or a deep neural-net), and the decision threshold b. The former provides a score which defines a ranking over examples, while the latter defines a decision boundary on the score that separates examples that are predicted to be relevant (positive) from those that are not.

Definition 3.2. The precision $P(f_b)$ and recall $R(f_b)$ of a classification rule are defined by:

$$P(f_b) = \frac{tp(f_b)}{tp(f_b) + fp(f_b)}$$

$$R(f_b) = \frac{tp(f_b)}{tp(f_b) + fn(f_b)} = \frac{tp(f_b)}{|Y^+|}$$

where tp, fp, fn are the true-positives, false-positives, and false-negative counts (respectively):

$$tp(f_b) = \sum_{i \in Y^+} \mathbb{1}_{f(x_i) \geq b} \qquad fp(f_b) = \sum_{i \in Y^-} \mathbb{1}_{f(x_i) \geq b}$$

We lower bound tp and upper bound fp by first writing

them in terms of the zero-one loss:

$$
\begin{aligned}
tp(f_b) &= \sum_{i \in Y^+} 1 - \ell_{01}(f_b, x_i, y_i) \\
fp(f_b) &= \sum_{i \in Y^-} \ell_{01}(f_b, x_i, y_i)
\end{aligned}
\tag{1}
$$

we do not need to bound fn because it shows up only in the denominator of the expression for recall and can be eliminated via $|Y^+| = tp + fn$. Now it is natural to bound these quantities by using a surrogate for the zero-one loss function such as the hinge loss:

$$
\begin{aligned}
tp_l(f_b) &\triangleq \sum_{i \in Y^+} 1 - \ell_h(f_b, x_i, y_i) \leq tp(f_b), \\
fp^u(f_b) &\triangleq \sum_{i \in Y^-} \ell_h(f_b, x_i, y_i) \geq fp(f_b),
\end{aligned}
\tag{2}
$$

where

$$\ell_h(f_b, x, y) \triangleq \max(0, 1 - y(f(x) - b))$$

is the hinge loss of the score $f(x) - b$ on point x with label $y \in \{-1, 1\}$. The right-hand inequalities follow directly. We note that in what follows we use the hinge-loss for simplicity but other losses could be used in all the results presented below (with the exception of the linear-fractional transformation of the F_β score in section 6). In the case of convex surrogates for the zero-one loss such as the log-loss or the smooth-hinge-loss [21] we get convex (and smooth) optimization problems. However, our method can also be used with non-convex surrogates such as the ramp-loss.

These simple bounds will allow us to bound a variety of global non-decomposable ranking measures including the **AUCROC**, **AUCPR**, F_β, etc.

4 Maximizing Recall at Fixed Precision

In this section we show how the building block bounds of (2) can provide a concave lower bound on the objective of maximum recall with *at least* α precision. A similar derivation could also be used to provide a bound on maximum precision given a minimum desired recall. Aside from the stand-alone usefulness of the **P@R** and **R@P** metrics, the developments here will underlie the construction for optimizing the maximum **AUCPR** objective that we present in the next section.

We begin by defining the maximum recall at fixed minimum precision problem:

$$
\begin{aligned}
\mathbf{R@P}\alpha = \quad &\max_{f} \quad R(f) \\
&s.t. \quad P(f) \geq \alpha
\end{aligned}
\tag{3}
$$

The above is a difficult combinatorial problem. Thus, instead of solving it directly, we optimize a lower bound similarly to how the hinge loss is used as a surrogate for accuracy in SVM optimization [8]. To do so, we write (3) as

$$\max_{f,b} \quad \frac{1}{|Y^+|} tp(f)$$
$$s.t. \quad tp(f) \geq \alpha(tp(f) + fp(f)).$$

To turn this objective into a tractable optimization surrogate, we use (2) to lower bound tp and upper bound fp:

$$\overline{\mathbf{R@P}\alpha} = \quad \max_{f,b} \quad \frac{1}{|Y^+|} tp_l(f)$$
$$s.t. \quad (1-\alpha)tp_l(f) \geq \alpha fp^u(f). \tag{4}$$

Lemma 4.1. *The relaxed problem $\overline{\mathbf{R@P}\alpha}$ is a concave lower bound for $\mathbf{R@P}\alpha$.*

Proof. To see that the surrogate problem is a lower bound of the original problem we notice that the surrogate recall $\frac{1}{|Y^+|} tp_l(f)$ is a lower bound on the true recall. In addition we notice that the surrogate precision $\overline{P} = \frac{tp_l(f)}{tp_l(f)+fp^u(f)}$ is a lower bound on the actual precision. Hence the feasible set of $\overline{\mathbf{R@P}\alpha}$ is contained in the feasible set of $\mathbf{R@P}\alpha$, as

$$P(f) \geq \overline{P}(f) \geq \alpha.$$

This proves that the surrogate problem is a lower bound on the original problem.

Finally, the objective of $\overline{\mathbf{R@P}\alpha}$ is concave, and the constraints are convex (in fact they are piece-wise linear). $\square$

The relaxed objective of (4) is now amenable to efficient optimization. To see this, we plug the explicit forms of $tp_l(f)$ and $fp^u(f)$ into (4):

$$\overline{\mathbf{R@P}\alpha} = \max_f \quad 1 - \frac{\mathscr{L}^+(f)}{|Y^+|}$$
$$s.t. \ (1-\alpha)(|Y^+| - \mathscr{L}^+(f)) \geq \alpha \mathscr{L}^-(f).$$

Where we use as a shorthand

$$\mathscr{L}^+(f) = \sum_{i \in Y^+} \ell_h(f, x_i, y_i)$$

for the loss on the positive examples, and similarly $\mathscr{L}^-$ is the sum of errors on the negative examples. We omit the explicit dependence on f when it is clear from context. Next, we rewrite the constraint and ignore the constant multiplier in the objective to obtain the

following equivalent (with respect to the optimal f, b) problem:

$$\min_f \mathscr{L}^+$$
$$s.t. \ \alpha \mathscr{L}^- + (1-\alpha)\mathscr{L}^+ \leq (1-\alpha)|Y^+|. \tag{5}$$

Now, applying Lagrange multiplier theory, we can equivalently consider the following objective:

$$\min_f \max_{\lambda \geq 0} \quad \mathscr{L}^+ + \lambda \left(\frac{\alpha}{1-\alpha} \mathscr{L}^- + \mathscr{L}^+ - |Y^+| \right).$$

Finally, after some regrouping of terms, this can be written as:

$$\min_f \max_{\lambda \geq 0} \quad (1+\lambda)\mathscr{L}^+ + \lambda \frac{\alpha}{1-\alpha} \mathscr{L}^- - \lambda|Y^+|. \tag{6}$$

We now face a saddle point problem, which we optimize using the following straightforward iterative stochastic gradient descent (SGD) updates:

$$f^{(t+1)} = f^{(t)} - \gamma \nabla L(f^{(t)}, \lambda^{(t)})$$
$$\lambda^{(t+1)} = \lambda^{(t)} + \gamma \nabla L(f^{(t+1)}, \lambda^{(t)})$$

where

$$L(f, \lambda) = (1+\lambda)\mathscr{L}^+(f) + \lambda \frac{\alpha}{1-\alpha} \mathscr{L}^-(f) - \lambda|Y^+|.$$

Lemma 4.2. *The above procedure converges to a fixed point if both $\mathscr{L}^+, \mathscr{L}^-$ are convex.*

The proof is straightforward can be found in [17, Section 3].

Aside from the obvious appeal of algorithmic simplicity, it is interesting to note that the above objective supports the standard practice of trying to achieve good **P@R** or **R@P** via example re-weighting. To see this, note that for a fixed λ, the minimization over f in Equation 6 is just a $c(\alpha, \lambda)$ weighted SVM. Specifically, after adding a regularization term, the SVM objective takes the form

$$\min_f \quad \sum_i \ell_h^c(f, x_i, y_i) + \|f\|^2, \tag{7}$$

where ℓ_h^c is the loss when a positive instance is weighted by $c(\alpha, \lambda) = \frac{(1+\lambda)(1-\alpha)}{\lambda\alpha}$. Because $c(\alpha, \lambda)$ is monotonic in λ, the problem can also be solved via a binary search for this single dual parameter.

4.1 Maximizing P@R

Following the same steps we can reach the following optimization problem:

$$\overline{\mathbf{P@R}}\beta = \min_f \max_{\lambda \geq 0} \frac{-|Y^+|\beta}{|Y^+|\beta + fp^u(f)} - \lambda \left(\frac{tp_l(f)}{|Y^+|} - \beta \right) \tag{8}$$

which seems odd at first as we expect it also to result in a re-weighting of the positives and negatives (mediated by λ) as in **R@P**. However, an equivalent problem which minimizes $1/P$ rather than $-P$ achieves the expected formulation

$$\overline{\textbf{P@R}}\beta = \min_f \max_\lambda \mathscr{L}^- + \lambda \left(\beta + \frac{\mathscr{L}^+}{|Y^+|} - 1 \right). \quad (9)$$

We note that this problem has the same minimizer as Equation 8, but not the same value.

5   Maximizing AUCPR

We are now ready to use our derivation of the R@P optimization objective in the previous section order to construct a concave lower-bound surrogate for **AUCPR**. A similar derivation could be used for **AUCROC** optimization.

To start, recall that **AUCPR** is simply an integral over **R@P** (equivalently **P@R**) values. That is:

$$\textbf{AUCPR}(f) = \max_f \int_\pi^1 R@P\alpha(f)d\alpha, \quad (10)$$

where π is the positive class prior, and $R@P\alpha(f)$ denotes the recall we achieve when using f as a score function with $b = b(\alpha)$ is a threshold which achieves precision α. Another way to think of b is via the optimization problem

$$R@P\alpha(f) = \max_b R(f_b)$$
$$\text{s.t. } P(f_b) = \alpha.$$

To apply our bounds to the objective of maximizing **AUCPR**(f), we first approximate the integral in Equation 10 by a discrete sum over a set of precision anchor values $A = \{\pi = \alpha_0 < \alpha_1 < \alpha_2 < \ldots < \alpha_k\}$:

$$\max_f \textbf{AUCPR}(f) =$$
$$\max_f \sum_{t=1}^k \Delta_t \left[\max_{b_t} R(f_{b_t}) \quad \text{s.t. } P(f_{b_t}) \geq \alpha_t \right], \quad (11)$$

where

$$\Delta_t = \alpha_t - \alpha_{t-1} \qquad \forall t = 1 \ldots k.$$

Naturally, one could take uniformly spaced α and set $\alpha_t = \pi + \frac{(1-\pi)t}{k}$, though this is not required.

Next, using the same technique we used for the maximum **R@P** objective, we relax the building block statistics and, after some algebraic manipulations and

application of the Lagrange multiplier theory, we get:

$$\min_{f,b_1,\ldots b_k} \max_{\lambda_1 \ldots \lambda_k} \sum_{t=1}^k \Delta_t \Big((1 + \lambda_t)\mathscr{L}^+(f, b_t)$$
$$+ \lambda_t \frac{\alpha_t}{1 - \alpha_t} \mathscr{L}^-(f, b_t) - \lambda_t |Y^+| \Big). \quad (12)$$

As before, we can solve this saddle point problem by SGD [17].

By replacing **R@P** with true positive rate at fixed false positive rate in the derivation above, we obtain a similar algorithm for optimizing **AUCROC**. The building block bounds from Section 3 can be used to generate surrogate objectives for true positive rate at the false positive rate anchor points.

An important consequence of the above derivation is that we can just as easily optimize for **AUCROC** and **AUCPR** in some limited range, e.g. for precision greater than some desired threshold. This would amount to constraining the range of precision anchor values in the above development, and can be optimized just as easily.

6   Optimizing the F_β Measure

To demonstrate the flexibility of our unifying framework, we now show that the building block bounds of (2), with a few additional manipulations, can also be used to optimize the commonly used F_β score. This score is a measure of the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision [26]. The F_β score is defined as:

$$F_\beta = (1 + \beta^2)\frac{P \cdot R}{\beta^2 P + R}$$

On the surface it is not clear how to use bounds on the tp and fp to bound F_β since these statistics appear both in the denominator and numerator. We can do so by first rewriting the F_β score in the well known type I and type II error form:

$$F_\beta = (1 + \beta^2)\frac{tp}{(1 + \beta^2)tp + \beta^2 fn + fp}$$
$$= (1 + \beta^2)\frac{tp}{\beta^2|Y^+| + tp + fp}$$

We now plug in the bounds from Equation 2, and get a surrogate function for F_β:

$$\overline{F_\beta} = (1 + \beta^2)\frac{tp_l}{\beta^2|Y^+| + tp_l + fp^u} \leq F_\beta.$$

The lower bound follows from the fact that subtracting the same quantity (the difference between tp and tp_l)

from both the numerator and denominator decreases the quotient, and that increasing the denominator (by replacing fp with fp^u) also decreases it.

Our goal now is to maximize the above lower bound efficiently. For simplicity we demonstrate this for F_1 but the details are essentially the same for F_β. First, we note that maximizing $\overline{F_1}$ is equivalent to minimizing $(\overline{F_1})^{-1}$, and write this objective as a fractional linear program [3]:

$$\min_{f,t,w} \quad \frac{|Y^+| + \sum_{i \in Y^+} t_i + \sum_{i \in Y^-} f_i}{\sum_{\forall i \in Y^+} t_i}$$

$$s.t.$$

$$\forall i \in Y^+ \quad t_i \leq 1, \quad t_i \leq w \cdot x_i$$

$$\forall i \in Y^- \quad f_i \geq 0, \quad f_i \geq 1 + w \cdot x_i.$$

We can now use the linear-fraction transformation [3] to derive the equivalent problem in variables τ, ϕ, ω and ϵ:

$$\min_{\phi,\tau,\omega,\epsilon} \quad |Y^+|\epsilon + \sum_{i \in Y^+} \tau_i + \sum_{i \in Y^-} \phi_i$$

$$s.t.$$

$$\forall i \in Y^+ \quad \tau_i \leq \epsilon, \quad \tau_i \leq \omega \cdot x_i$$

$$\forall i \in Y^- \quad \phi_i \geq 0, \quad \phi_i \geq \epsilon + \omega \cdot x_i \qquad (13)$$

$$\sum_{i \in Y^+} \tau_i = 1$$

$$\epsilon \geq 0,$$

where we used the mappings $\omega = \epsilon w$, $\phi_i = \epsilon f_i$, $\tau_i = \epsilon t_i$ and $\epsilon = \frac{1}{\sum_i t_i}$. The resulting linear program can be of course solved in various ways, e.g. using an iterative gradient ascent procedure as with the **R@P** and **AUCPR** objectives.

Alternatively, the task of maximizing the F_β measure can be solved using a constrained optimization approach. First we write the minimization of $(F_1)^{-1}$ task as a function of $\mathscr{L}^+, \mathscr{L}^-$:

$$\min(F_1)^{-1} = \min_f \frac{(|Y^+| + |Y^+| - \mathscr{L}^+ + \mathscr{L}^-)}{(|Y^+| - \mathscr{L}^+)}, \quad (14)$$

which is equivalent to

$$\min_{f,\psi} \frac{(|Y^+| + \psi + \mathscr{L}^-)}{\psi}$$

$$s.t. \ \psi = |Y^+| - \mathscr{L}^+ \qquad (15)$$

after defining an auxiliary variable $\psi = (|Y^+| - \mathscr{L}^+)$. Some simple algebra shows that the above is equivalent to

$$\min_{f,\psi} \max_\lambda \psi^{-1}\mathscr{L}^- + \lambda\mathscr{L}^+ + (\psi^{-1} - \lambda)|Y^+| + \psi\lambda.$$

$$(16)$$

From this formulation we see again that given ψ, λ we have a weighted classification problem as was proved by a different technique in [18].

7 Experimental Evaluation

In this section we demonstrate the merit of our approach for learning with a non-decomposable objective. We focus mostly on the optimization of the **AUCPR** objective due to its wide popularity in ranking scenarios and, as discussed, the fact that existing methods for optimizing this metric are not sufficiently scalable. With this in mind we consider three challenging problems, two of which are substantially (by orders of magnitude) larger than the those considered in the literature (e.g. [23]).

7.1 CIFAR-10

The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. The goal is to disinguish between the 10 classes. There are 50000 training images and 10000 test images. As our baseline we use the deep convolutional network from TensorFlow.org. Both the baseline model (trained with a soft-max loss) and our model (trained with the **AUCPR** loss) were optimized for 200,000 SGD steps with 128 images per batch. All the models were trained on a single tesla k40 GPU for about eight hours. Recall that our method relies on K discrete anchor points to approximate the **AUCPR** integral. To evaluate the robustness for this choice, we learned three models with 5, 10 and 20 points. Results were essentially identical for all of these settings, so for brevity only results with $K = 10$ are reported below. We also compared to the standard pairwise **AUCROC** surrogate [20].

The advantage of optimizing for the objective of interest is clear: optimizing for **AUCPR** rather than accuracy increases the **AUCPR** metric from 84.6% to 94.2%. Results for other metrics are presented below in Table 1.

To get a more refined view of the differences between the baseline model and our approach, Figure 2 shows the aggregate precision-recall curve across all classes (left), and the per-class breakdown (right). The aggregate difference between the two models is evident and the advantage of our method in the interesting range of high precision is particularly impressive. Looking at the per-class performance, we see that our approach

improves the **AUCPR** for *all* 10 classes, and substantially so for classes where the performance of the baseline is poor.

7.2 ImageNet

The ILSVRC 2012 image dataset [22] contains 1.2 million images for training and 50K for validation. The goal is to distinguish between 1000 object classes. This dataset, also known as ImageNet, is used as the basis for competitions where the accuracy at the top 1 or 5 predictions is measured. We used the same dataset to demonstrate that we can trade off accuracy and **AUCPR** at scale. We use the Inception-v3 open source implementation from TensorFlow.org [1] as our baseline. The ImageNet experiments were trained with 50 tesla K40 GPU replicas for three days performing about 5M mini-batch updates. Using the same architecture, we optimize for **AUCPR**, allowing both the baseline and our method the same training time. By using $K = 5$ anchor points to approximate the **AUCPR** integral, we increase the **AUCPR** from 82.2% for the baseline to 83.3%, while decreasing accuracy by 0.4%. We note that, generally speaking, improvements on the order of 1% for ImageNet are considered substantial.

7.3 JFT

To demonstrate merit and applicability of our method on a truly large-scale domain, we consider the Google internal JFT dataset. This dataset has over 300 million labeled images with about $20,000$ labels. As our baseline we use a deep convolutional neural network which is based on the Inception architecture [24]. The specific architecture used is Google's current state-of-the-art model. Performance of models on this data is evaluated first and foremost using the **AUCPR** metric, yet the baseline model is trained to maximize accuracy via a logistic loss function. To learn a model using our approach, we start

Metric \Gain over	**AUCROC**	softmax
AUCPR	0.0%	9.6%
P@R70	0.2%	13.5%
P@R95	0.8%	24.1%
R@P70	-0.2%	12.7%
R@P95	1.9%	36.6%

Table 1: The gain over a baseline loss (in absolute percentage points) in various metrics when optimizing for that metric with our framework. The baseline losses considered are pairwise **AUCROC** and standard softmax cross-entropy. In all cases, the model architectures being optimized are identical.

training from the pre-trained parameters (training from scratch is a multi-month process), and optimize the **AUCPR** objective for several days. To have a fair comparison, we also allow the baseline model to continue training for the same amount of time. While the baseline model achieves an **AUCPR** of 42%, the model optimized with our surrogate achieves an **AUCPR** of 48%, a substantial improvement.

8 Summary and Future Directions

In this work we addressed the challenge of scalable optimization non-decomposable ranking-based objective. We introduced simple building block bounds that provide a unified framework for efficient optimization of a wide range of such objectives. We demonstrated the empirical effectiveness of our approach on several real-life datasets.

Importantly, some of the problems we consider are dramatically larger than those previously considered in the literature. In fact, our approach is essentially as efficient as optimization of the fully decomposable accuracy loss. Indeed, our method can be coupled with any shallow (SVM, logistic regression) or deep (CNN) architecture with negligible cost on performance.

Aside from the obvious appeal of scalability, our unified approach also opens the door for novel optimization of more refined objectives. For example, maximizing the area under the ROC curve for a pre-specified range of the false-positive rate is as easy as maximizing the area under the precision recall curve altogether. In future work, we plan to explore the importance of such flexibility for real-world ranking problems.

References

[1] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Stephen Boyd, Corinna Cortes, Mehryar Mohri, and Ana Radovanovic. Accuracy at the top. In *Advances in neural information processing systems*, pages 953–961, 2012.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international*
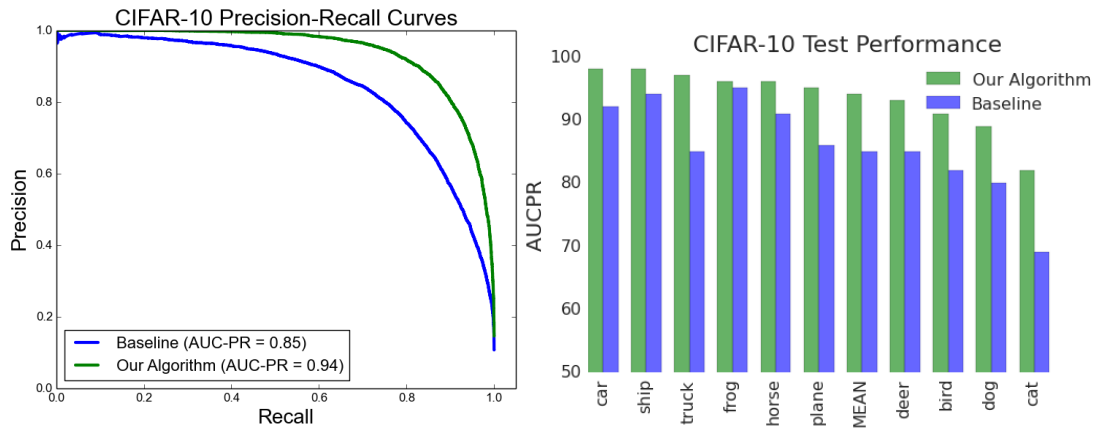
Figure 2: Comparison of a baseline model trained to optimize accuracy and a model trained to optimize **AUCPR** using our method on the CIFAR10 dataset. (left) Shows the aggregate precision-recall curve for all 10 classes. (right) Compares the **AUCPR** for each of the 10 classes.

conference on Machine learning, pages 161–168. ACM, 2006.

[5] Zachary Chase Lipton, Charles Elkan, and Balakrishnan Narayanaswamy. Thresholding classifiers to maximize f1 score. *arXiv preprint arXiv:1402.1892*, 2014.

[6] Yunmei Chen, Guanghui Lan, and Yuyuan Ouyang. Optimal primal-dual methods for a class of saddle point problems. *SIAM Journal on Optimization*, 24(4):1779–1814, 2014.

[7] Corinna Cortes and Mehryar Mohri. Auc optimization vs. error rate minimization. *Advances in neural information processing systems*, 16(16):313–320, 2004.

[8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[9] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[10] Alan Herschtal and Bhavani Raskutti. Optimising area under the roc curve using gradient descent. In *Proceedings of the twenty-first international conference on Machine learning*, page 49. ACM, 2004.

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

[12] Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Online and stochastic gradient methods for non-decomposable loss functions. In *Advances in Neural Information Processing Systems*, pages 694–702, 2014.

[13] Donald Metzler and W Bruce Croft. A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479. ACM, 2005.

[14] Donald A Metzler, W Bruce Croft, and Andrew McCallum. Direct maximization of rank-based metrics for information retrieval. *unpublished*, 2005.

[15] Pritish Mohapatra, CV Jawahar, and M Pawan Kumar. Efficient optimization for average precision svm. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2014.

[16] Ye Nan, Kian Ming Chai, Wee Sun Lee, and Hai Leong Chieu. Optimizing f-measure: A tale of two approaches. *arXiv preprint arXiv:1206.4625*, 2012.

[17] Angelia Nedić and Asuman Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009.

[18] Shameem Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. Optimizing f-measures by cost-sensitive classification. In *Advances in Neural Information Processing Systems*, pages 2123–2131, 2014.

[19] C Quoc and Viet Le. Learning to rank with nonsmooth cost functions. *Proceedings of the Advances in Neural Information Processing Systems*, 19:193–200, 2007.

[20] Alain Rakotomamonjy. Optimizing area under roc curve with svms. In *ROCAI*, pages 71–80, 2004.

[21] Jason DM Rennie. Smooth hinge classification. *Proceeding of Massachusetts Institute of Technology*, 2005.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[23] Yang Song, Alexander G Schwing, Richard S Zemel, and Raquel Urtasun. Direct loss minimization for training deep neural nets. *arXiv preprint arXiv:1511.06411*, 2015.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[25] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. In *Journal of Machine Learning Research*, pages 1453–1484, 2005.

[26] CJ Van Rijsbergen. Information retrieval. dept. of computer science, university of glasgow. *URL: citeseer. ist. psu. edu/vanrijsbergen79information. html*, 1979.

[27] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278. ACM, 2007.