Active Testing: An Efficient and Robust Framework for Estimating Accuracy.

Phuc Nguyen [1] **Deva Ramanan** [2] **Charless Fowlkes** [1]

Abstract

Much recent work on visual recognition aims to scale up learning to massive, noisily-annotated datasets. We address the problem of scaling-up the *evaluation* of such models to large-scale datasets with noisy labels. Current protocols for doing so require a human user to either vet (re-annotate) a small fraction of the test set and ignore the rest, or else correct errors in annotation as they are found through manual inspection of results. In this work, we re-formulate the problem as one of *active testing*, and examine strategies for efficiently querying a user so as to obtain an accurate performance estimate with minimal vetting. We demonstrate the effectiveness of our proposed active testing framework on estimating two performance metrics, Precision@K and mean Average Precision, for two popular computer vision tasks, multi-label classification and instance segmentation. We further show that our approach is able to save significant human annotation effort and is more robust than alternative evaluation protocols.

1. Introduction

Visual recognition is undergoing a period of transformative progress, due in large part to the success of deep architectures trained on massive datasets with supervision. While visual data is in ready supply, high-quality supervised labels are not. One attractive solution is the exploration of unsupervised learning. However, regardless how they are trained, one still needs to *evaluate* accuracy of the resulting systems. Given the importance of rigorous, empirical benchmarking, it appears impossible to avoid the costs of assembling high-quality, human-annotated test data for test evaluation.

Unfortunately, manually annotating ground-truth for large-scale test datasets is often prohibitively expensive, particularly for rich annotations required to evaluate object detection and segmentation. Even simple image tag annotations pose an incredible cost at scale [1]. In contrast, obtaining noisy or partial annotations is often far cheaper or even free. For example, numerous social media platforms produce image and video data that are dynamically annotated with user-provided tags (Flickr, Vine, Snapchat, Facebook, YouTube). While much work has explored the use of such massively-large "webly-supervised" data sources for learning (Wu et al., 2015; Yu et al., 2014; Li et al., 2017; Veit et al., 2017), we instead focus on them for evaluation.

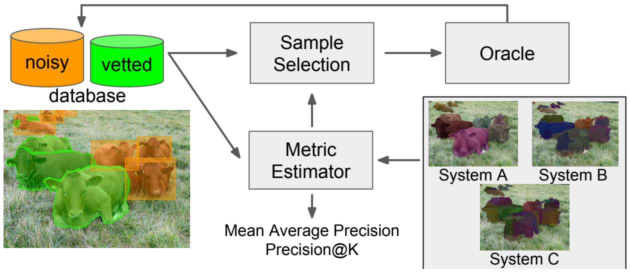


Figure 1. Classic methods for benchmarking algorithm performance require a test-set with high-quality labels. While it is often easy to obtain large-scale data with noisy labels, test evaluation is typically carried out on only a small fraction of the data that has been manually cleaned-up (or "vetted"). We show that one can obtain dramatically more accurate estimates of performance by using the vetted-set to train a statistical estimator that both (1) reports improved estimates and (2) actively selects the next batch of test data to vet. We demonstrate that such an "active-testing" process can efficiently benchmark performance and and rank visual recognition algorithms.

How can we exploit such partial or noisy labels during testing? With a limited budget for vetting noisy ground-truth labels, one may be tempted to simply evaluate performance on a small set of clean data, or alternately just trust the cheap-but-noisy labels on the whole dataset. However, such approaches can easily give an inaccurate impression of system performance. We show in our experiments that these naive approaches can produce alarmingly-incorrect estimates of comparative model performance. Even with a significant fraction of vetted data, naive performance esti-

[1]University of California, Irvine [2]Carnegie Mellon University. Correspondence to: Phuc Nguyen <nguyenpx@uci.edu>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

[1] For example, NUS-WIDE, (Chua et al., 2009) estimated 3000 man-hours to semi-manually annotate a relatively small set of 81 concepts across 270K images

mates can incorrectly rank two algorithms in 15% of trials, while our active testing approach significantly reduces this misranking error to 3%.

The problem of label noise even exists for "expertly" annotated datasets, whose construction involves manual selection of a test set which is deemed representative in combination with crowd-sourced labeling by multiple experts (Rashtchian et al., 2010; Khattak & Salleb-Aouissi, 2011). Preserving annotation quality is an area of intense research within the HCI/crowdsourcing community (Kamar et al., 2012; Sheshadri & Lease, 2013). In practice, annotation errors are often corrected incrementally through multiple rounds of interactive error discovery and visual inspection of algorithm test results over the lifetime of the dataset. For example, in evaluating object detectors, the careful examination of detector errors on the test set (Hoiem et al., 2012) often reveals missing annotations in widely-used benchmarks (Lin et al., 2014; Everingham et al., 2015; Dollar et al., 2012) and may in turn invoke further iterations of manual corrections (e.g., (Mathias et al., 2014)). In this work, we formalize such ad-hoc practices in a framework we term *active testing*, and show that significantly improved estimates of accuracy can be made through simple statistical models and active annotation strategies.

2. Related Work

Benchmarking: Empirical benchmarking is now widely considered to be an integral tool in the development of vision and learning algorithms. Rigorous evaluation, often in terms of challenge competitions (Russakovsky et al., 2015; Everingham et al., 2010) on held-out data, serves to formally codify proxies for scientific or application goals and provides quantitative ways to characterize progress towards them. The importance and difficulties of test dataset construction and annotation are now readily appreciated (Ponce et al., 2006; Torralba & Efros, 2011).

Benchmark evaluation can be framed in terms of the well-known *empirical risk* minimization approach to learning (Vapnik, 1992). Benchmarking seeks to estimate the risk, defined as the expected loss of an algorithm under the true data distribution. Since the true distribution is unknown, the expected risk is estimated by computing loss a finite sized sample test set. Traditional losses (such as 0-1 error) decompose over test examples, but we are often interested in multivariate ranking-based metrics that do not decompose (such as Precision@K and Average Precision (Joachims, 2005)). Defining and estimating expected risk for such metrics is more involved (e.g., Precision@K should be replaced by precision at a specified quantile (Boyd et al., 2012)) but generalization bounds are known (Agarwal et al., 2005; Hill et al., 2002). For simplicity, we focus on the problem of estimating the empirical risk on a fixed, large but finite test set.

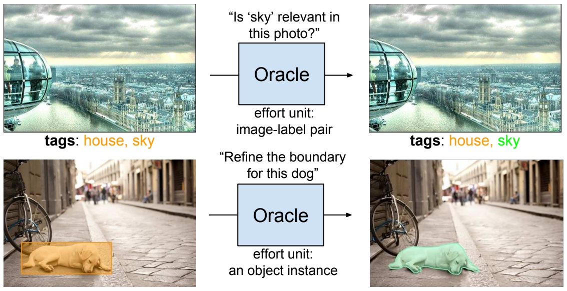

"Is 'sky' relevant in this photo?"
Oracle
effort unit: image-label pair
tags: house, sky
tags: house, sky
"Refine the boundary for this dog"
Oracle
effort unit: an object instance

Figure 2. Vetting Procedure. The figure shows the vetting procedures for the multi-label classification (top) and instance segmentation (bottom) tasks. The annotations on the left are often incomplete and noisy, but significantly easier to obtain. These initial noisy annotations are "vetted" and corrected if necessary by a human. We quantify human effort in units of the number of image-label pairs corrected or object segment masks specified.

Semi-supervised testing: To our knowledge, there have only been a handful of works specifically studying the problem of estimating recognition performance on partially labeled test data. Anirudh et al. (Anirudh & Turaga, 2014) study the problem of 'test-driving' a detector to allow the users to get a quick sense of the generalizability of the system. Closer to our approach is that of Welinder et al. (Welinder et al., 2013), who estimate the performance curves using a generative model for the classifier's confidence scores. Their approach leverages ideas from the semi-supervised learning literature while our approach builds on active learning.

The problem of estimating benchmark performance from sampled relevance labels has been explored more extensively in the information retrieval literature where complete annotation was acknowledged as infeasible. Initial work focused on deriving labeling strategies that produce low-variance and unbiased estimates (Yilmaz & Aslam, 2006; Aslam et al., 2006) and identifying performant retrieval systems (Moffat et al., 2007). (Sabharwal & Sedghi, 2017) give error bounds for estimating PR and ROC curves by choosing samples to label based on the system output ranking. (Gao et al., 2014) estimate performance using an EM algorithm to integrate relevance judgements. (Li & Kanoulas, 2017) and (Rahman et al., 2018) take a strategy similar to ours in actively selecting test items to label as well as estimating performance on remaining unlabeled data.

Active learning: Our proposed formulation of active testing is closely related to active learning. From a theoretical perspective, active learning can provide strong guarantees of efficiency under certain restrictions (Balcan & Urner, 2016). Human-in-the-loop active learning approaches have been well explored for addressing training data collection in visual recognition systems (Branson et al., 2010; Wah et al., 2011; Vijayanarasimhan & Grauman, 2014). One can view *active testing* as a form of active learning where the actively-trained model is a statistical predictor of performance on a test set. Active learning is typically cast within the standard machine-learning paradigm, where the goal is to (interactively) learn a model that makes accurate per-example predictions on held-out *i.i.d* data. In this case, generalization is of paramount importance. On the other hand, active-testing interactively learns a model that makes *aggregate* statistical predictions over a *fixed* dataset. This means that models learned for active-testing (that say, predict average precision) need not generalize beyond the test set of interest. This suggests that one can be much more aggressive in overfitting to the statistics of the data at hand.

3. Framework for Active Testing

In this section, we introduce the general framework for active testing. Figure 1 depicts the overall flow of our approach. Our evaluation database initially contains test examples with inaccurate (noisy) annotations. We select a batch of data items whose labels will be manually vetted by an oracle (e.g., in-house annotators or a crowd-sourced platform such as Mechanical Turk). Figure 2 shows examples of such noisy labels and queries to Oracle. The evaluation database is then updated with these vetted labels to improve estimates of test performance. Active testing consists of two key components: a *metric estimator* that estimates model performance from test data with a mix of noisy and vetted labels, and a *vetting strategy* which selects the subset of test data to be labeled in order to achieve the best possible estimate of the true performance.

3.1. Performance Metric Estimators

We first consider active testing for a simple binary prediction problem and then extend this idea to more complex benchmarking tasks such as multi-label tag prediction and instance segmentation. As a running example, assume that we are evaluating an system that classifies an image (e.g., as containing a cat or not). The system returns of confidence score $s_i \in \mathbb{R}$ for each test example $i \in \{1 \dots N\}$. Let y_i denote a "noisy" binary label for example i (specifying if a cat is present), where the noise could arise from labeling the test set using some weak-but-cheap annotation technique (e.g., user-provided tags, search engine results, or approximate annotations). Finally, let z_i be the true latent binary label whose value can be obtained by rigorous human inspection of the test data item.

Typical benchmark performance metrics can be written as a function of the true ground-truth labels and system confidences. We focus on metrics that only depend on the rank ordering of the confidence scores and denote such a metric generically as $Q(\{z_i\})$ where for simplicity we hide the dependence on s by assuming that the indices are always *sorted* according to s_i so that $s_1 \geq \cdots \geq s_N$. For example, commonly-used metrics for binary labeling include precision@K and average precision (AP):

$$Prec@K(\{z_1, \dots, z_N\}) = \frac{1}{K} \sum_{i \leq K} z_i \tag{1}$$

$$AP(\{z_1, \dots, z_N\}) = \frac{1}{N_p} \sum_{k} \frac{z_k}{k} \sum_{i \leq k} z_i \tag{2}$$

where N_p is the number of positives. We include derivations in supplmental material.

Estimation with partially vetted data: In practice, not all the data in our test set will be vetted. Let us divide the test set into two components, the unvetted set U for which we only know the approximate noisy labels y_i and the vetted set V, for which we know the ground-truth label. With a slight abuse of notation, we henceforth treat the true label z_i as a random variable, and denote its observed realization (on the vetted set) as $\tilde{z}_i$. The simplest strategy for estimating the true performance is to ignore unvetted data and only measure performance Q on the vetted subset:

$$Q(\{\tilde{z}_i : i \in V\}) \qquad \textbf{[Vetted Only]} \tag{3}$$

This represents the traditional approach to empirical evaluation in which we collect a single, vetted test dataset and ignore other available test data. This has the advantage that it is unbiased and converges to the true empirical performance as the whole dataset is vetted. The limitation is that it makes use of only fully-vetted data and the variance in the estimate can be quite large when the vetting budget is limited.

A natural alternative is to incorporate the unvetted examples by simply substituting y_i as a "best guess" of the true z_i. We specify this *naive* assumption in terms of a distribution over all labels $z = \{z_1, \dots, z_N\}$:

$$p_{naive}(z) = \prod_{i \in U} \delta(z_i = y_i) \prod_{i \in V} \delta(z_i = \tilde{z}_i) \tag{4}$$

where $\tilde{z}_i$ is the label assigned during vetting. Under this assumption we can then compute an expected benchmark performance:

$$E_{p_{naive}(z)}\Big[Q(z)\Big] \qquad \textbf{[Naive Estimator]} \tag{5}$$

which amounts to simply substituting $\tilde{z}_i$ for vetted examples and y_i for unvetted examples.

Unfortunately, the above performance estimate may be greatly affected by noise in the nosiy labels y_i. For example, if there are systematic biases in the y_i, the performance estimate will similarly be biased. We also consider more general scenarios where side information such as features

Algorithm 1 Active Testing Algorithm

```
Input: unvetted set $U$, vetted set $V$, total budget $T$, vetting strategy $VS$, system scores $S = \{s_i\}$, estimator $p_{est}(z)$
while $T \geq 0$ do
    Select a batch $B \subseteq U$ according to vetting strategy $VS$.
    Query oracle to vet $B$ and obtain true annotations $\tilde{z}$.
    $U = U \setminus B, V = V \cup B$
    $T = T - |B|$
    Fit estimator $p_{est}$ using $U, V, S$.
end while
Estimate performance using $p_{est}(z)$
```

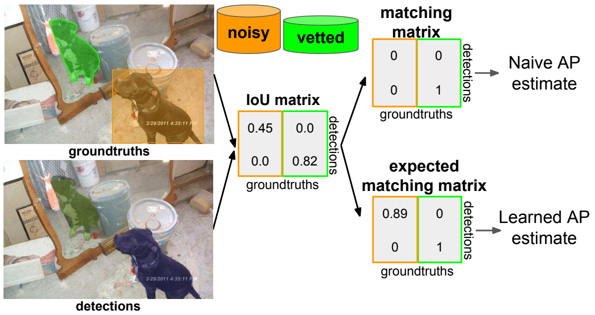


Figure 3. Standard instance segmentation benchmarks ignore unvetted data (top pathway) when computing Average Precision. Our proposed estimator for this task computes an expected probability of a match for coarse bounding box annotations when vetted instance masks aren't available.

of the test items and distribution of scores of the classifier under test may also be informative. We thus propose computing the expected performance under a more sophisticated estimator:

$$p_{est}(z) = \prod_{i \in U} p(z_i|\mathcal{O}) \prod_{i \in V} \delta(z_i = \tilde{z}_i) \tag{6}$$

where $\mathcal{O}$ is the total set of all observations available to the benchmark system (e.g. noisy labels, vetted labels, classifier scores, data features). We make the plausible assumption that the distribution of unvetted labels factors conditioned on $\mathcal{O}$.

Our proposed active testing framework (see Alg 1) estimates this distribution $p_{est}(z)$ based on available observations and predicts expected benchmark performance under this distribution:

$$E_{p_{est}(z)}\Big[Q(z)\Big] \qquad \textbf{[Learned Estimator]} \tag{7}$$

Computing expected performance: Given posterior estimates $p(z_i|\mathcal{O})$ we can always compute the expected performance metric Q by generating samples from these distributions, computing the metric for each joint sample, and average over samples. Here we introduce two applications (studied in our experiments) where the metric is linear or quadratic in z, allowing us to compute the expected performance in closed-form.

Multi-label Tags: Multi-label tag prediction is a common task in video/image retrieval. Following recent work (Joulin et al., 2016; Gong et al., 2013; Izadinia et al., 2015; Guillaumin et al., 2009), we measure accuracy with Precision@K - e.g., what fraction of the top K search results contain the tag of interest? In this setting, noisy labels y_i come from user provided tags which may contain errors and are typically incomplete. Conveniently, we can write expected performance Eq. 7 for Precision@K for a single tag in closed form:

$$E[Prec@K] = \frac{1}{K}\Big(\sum_{i \in V_K} \tilde{z}_i + \sum_{i \in U_K} p(z_i = 1|\mathcal{O})\Big) \tag{8}$$

where we write V_K and U_K to denote the vetted and unvetted subsets of K highest-scoring examples in the total set $V \cup U$. Some benchmarks compute an aggregate mean precision over all tags under consideration, but since this average is linear, one again obtains a closed form estimate.

Instance segmentation: Instance segmentation is another natural task for which to apply active testing. It is well known that human annotation is prohibitively expensive – (Cordts et al., 2016) reports that an average of more than 1.5 *hours* is required to annotate a single image. Widely used benchmarks such as (Cordts et al., 2016) release small fraction of images annotated with high quality, along with a larger set of noisy or "coarse"-quality annotations. Other instance segmentation datasets such as COCO (Lin et al., 2014) are constructed stage-wise by first creating a detection dataset which only indicates rectangular bounding boxes around each object which are subsequently refined into a precise instance segmentations. Fig. 3 shows an example of a partially vetted image in which some instances are only indicated by a bounding box (noisy), while others have a detailed mask (vetted).

When computing Average Precision, a predicted instance segmentation is considered a true positive if it has sufficient intersection-over-union (IoU) overlap with a ground-truth instance. In this setting, we let the variable z_i indicate that predicted instance i is matched to a ground-truth instance and has an above threshold overlap. Assuming independence of z_i's, the expected AP can be written as (see supplement for proof):

$$\begin{aligned} E[AP] = \frac{1}{N_p}\Big(&\sum_{k \in V} \tilde{z}_k E[Prec@k] \\ &+ \sum_{k \in U} p(z_k = 1|\mathcal{O}) E[Prec@k]\Big) \end{aligned} \tag{9}$$

In practice, standard instance segmentation benchmarks are somewhat more complicated. In particular, they enforce one-to-one matching between detections and ground-truth. For example, if two detections overlap a ground-truth instance, only one is counted as a true positive while the other

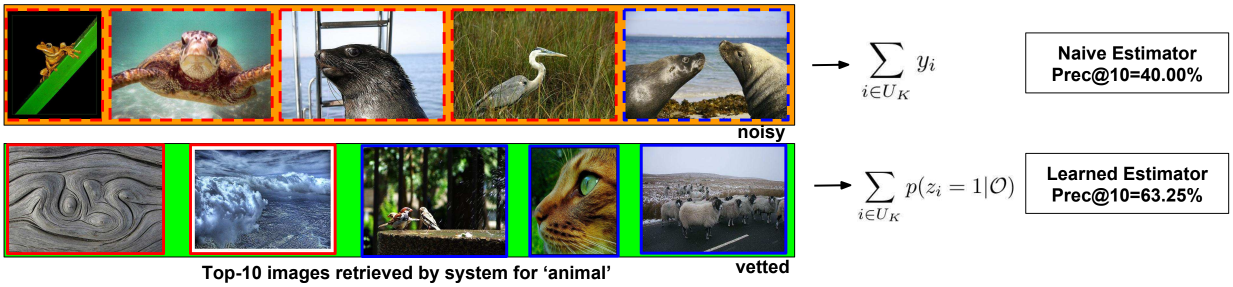


Figure 4. Estimating $Prec@K$. Images at left are the top K=10 entries returned by the system being evaluated. The image border denotes the current label and vetting status (solid blue/red = vetted positive/negative, and dotted blue/red = noisy positive/negative). Estimates of precision can be significantly improved by using a learned estimator trained on the statistics of examples that have already been vetted. Current approaches that evaluate on vetted-only or vetted+noisy labels (naive) produce poor estimates of precision (30% and 40% respectively). Our learned estimator is much closer to the true precision (63% vs 80% respectively).

is scored as a false positive. This also holds for multi-class detections - if a detection is labeled as a dog (by matching to a ground-truth dog), it can no longer be labeled as cat. While this interdependence can in principle be modeled by the conditioning variables $\mathcal{O}$ which could include information about which class detections overlap, in practice our estimators for $p(z_i = 1|\mathcal{O})$ do not take this into account. Nevertheless, we show that such estimators provide remarkably good estimates of performance.

Fitting estimators to partially vetted data: We alternate between vetting small batches of data and refitting the estimator to the vetted set. For multi-label tagging, we update estimates for the prior probability that a noisy tag for a particular category will be flipped when vetted $p(\tilde{z}_i \neq y_i)$. For instance segmentation, we train a per-category classifier that uses sizes of the predicted and unvetted ground-truth bounding box to predict whether a detected instance will overlap the ground-truth. We discuss the specifics of fitting these particular estimators in the experimental results.

3.2. Vetting Strategies

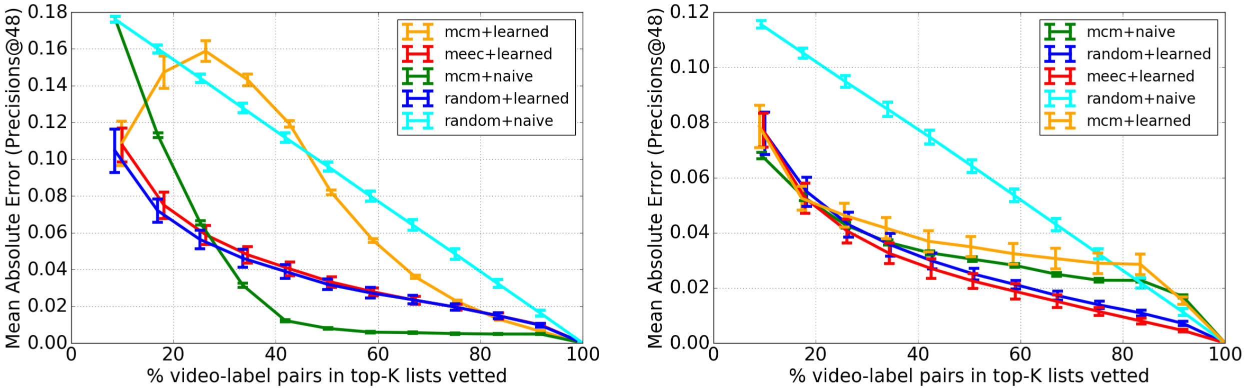
The second component of the active testing system is a strategy for choosing the "next" data samples to vet. The goal of such a strategy is to produce accurate estimates of benchmark performance with fewest number of vettings. An alternate, but closely related goal, is to determine the benchmark rankings of a set of recognition systems being compared. The success of a given strategy depends on the distribution of the data, the chosen estimator, and the system(s) under test. We consider several selection strategies, motivated by existing data collection practice and modeled after active learning, which adapt to these statistics in order to improve efficiency.

Random Sampling: The simplest vetting strategy is to choose test examples to vet at random. The distribution of examples across categories often follows a long-tail distribution. To achieve faster uniform convergence of performance estimates across all categories, we use a hierarchical sampling approach in which we first sample a category and then select a sub-batch of test examples to vet from that category. This mirrors the way, e.g. image classification and detection datasets are manually curated to assure a minimum number of examples per category.

Most-Confident Mistake (MCM): This strategy selects unvetted examples for which the system under test reports a high-confidence detection/classification score, but which are considered a mistake according to the current metric estimator. Specifically, we focus on the strategy of selecting *Most-confident Negative* which is applicable to image/video tagging where the set of user-provided tags are often incomplete. The intuition is that, if a high-performance system believes that the current sample is a positive with high probability, it's likely that the noisy label is at fault. This strategy is motivated by experience with object detection benchmarks where, e.g., visualizing high-confident false positive face detections often reveals missing annotations in the test set (Mathias et al., 2014).

Maximum Expected Estimator Change (MEEC): In addition to utilizing the confidence scores produced by the system under test, it is natural to also consider the uncertainty in the learned estimator $p_{est}(z)$. Exploiting the analogy of active testing with active learning, it is natural to vet samples that are most confusing to the current estimator (e.g., with largest entropy), or ones that will likely generate a large update to the estimator (e.g., largest information gain).

Specifically, we explore a active selection strategy based on *maximum expected model change* (Settles, 2010), which in our case corresponds to selecting a sample that yields the largest expected change in our estimate of Q. Let $E_{p(z|V)}[Q(z)]$ be the expected performance based on the distribution $p(z|V)$ estimated from the current vetted set V. $E_{p(z|V,z_i)}[Q(z)]$ be the expected performance after vetting example i and updating the estimator based on the outcome. The *actual* change in the estimate of Q depends on the



(a) Micro-Videos (b) NUS-WIDE

Figure 5. Results for multi-label classification task. The figures show the mean and standard deviation of the estimated Precision@48 at different amount of annotation efforts. Using a fairly simple estimator and vetting strategy, the proposed framework can estimate the performance very closely to the true values. For references, the precision@48 averaged across classes is 20.06% and 19.88% for Microvideos and NUS-WIDE respectively.

realization of the random variable z_i:

$$\Delta_i(z_i) = \left| E_{p(z|V,z_i)}[Q(z)] - E_{p(z|V)}[Q(z)] \right| \tag{10}$$

We can choose the example i with the largest *expected* change, using the current estimate of the distribution over $z_i \sim p(z_i|V)$ to compute the expected change $E_{p(z_i|V)}\left[\Delta_i(z_i)\right]$.

For $Prec@K$, this expected change is given by:

$$E_{p(z_i|V)}\left[\Delta_i(z_i)\right] = \frac{2}{K} p_i(1 - p_i) \tag{11}$$

where we write $p_i = p(z_i = 1|\mathcal{O})$. Interestingly, selecting the sample yielded the maximum expected change in the estimator corresponds to a standard *maximum entropy* selection criteria for active learning. Similarly, in the supplement we show that for AP:

$$E_{p(z_i|V)}\left[\Delta_i(z_i)\right] = \frac{1}{N_p} r_i p_i(1 - p_i) \tag{12}$$

where r_i is the proportion of unvetted examples scoring higher than example i. In this case, we select an example to vet which has high-entropy and for which there is a relatively small proportion of higher-scoring unvetted examples.

4. Experiments

We validate our active testing framework on two specific applications, multi-label classification and instance segmentation. For each of these applications, we describe the datasets and systems evaluated and the specifics of the estimators and vetting strategies used.

4.1. Active Testing for Multi-label Classification

NUS-WIDE: This dataset contains 269,648 Flickr images with 5018 unique tags. The authors also provide a 'semi-complete' ground-truth via manual annotations for 81 concepts. We removed images that are no longer available and images that doesn't contain one of the 81 tags. We are left with around 100K images spanning across 81 concepts. (Izadinia et al., 2015) analyzed the noisy and missing label statistics for this dataset. Given that the tag is relevant to the image, there is only 38% chance that it will appear in the noisy tag list. If the tag does not apply, there's 1% chance that it appears anyway. They posited that the missing tags are either non-entry level categories (e.g., person) or they are not important in the scene (e.g., clouds and buildings).

Micro-videos: Micro-videos have recently become a prevalent form of media on many social platforms, such as Vine, Instagram, and Snapchat. (Nguyen et al., 2016) formulated a multi-label video-retrieval/annotation task for a large collection of Vine videos. They introduce a micro-video dataset, **MV-85k** containing 260K videos with 58K tags. This dataset, however, only provides exhaustive vetting for a small subset of tags on a small subset of videos. We vetted 26K video-tag pairs from this dataset, spanning 17503 videos and 875 tags. Since tags provided by users have little constraints, this dataset suffers from both under-tagging and over-tagging. Under-tagging comes from not-yet popular concepts, while over-tagging comes from the spamming of extra tags In our experiments we use a subset of 75 tags.

Recognition systems: To obtain the classification results, we implement two multi-label classification algorithms for images (NUSWIDE) and videos (Microvideos). For NUS-WIDE, we trained a multi-label logistic regression model built on the pretrained ResNet-50 (He et al., 2016) features. For Micro-videos, we follow the state-of-the-art video action recognition framework (Wang et al., 2016) modified for the multi-label setting to use multiple logistic cross-entropy losses.

Learned Estimators: We use Precision@48 as a evaluation metric. For tagging, we estimate the posterior over unvetted tags, $p(z_i|\mathcal{O})$, based on two pieces of observed information: the statistics of noisy labels y_i on vetted examples, and the system confidence score, s_i. This posterior probability can be derived as (see supplement for proof):

$$p(z_i|s_i, y_i) = \frac{p(y_i|z_i)p(z_i|s_i)}{\sum_{v\in\{0,1\}} p(y_i|z_i = v)p(z_i = v|s_i)} \qquad (13)$$

Given some vetted data, we fit the tag-flipping priors $p(y_i|z_i)$ by standard maximum likelihood estimation (counting frequencies). The posterior probabilities of the true label given the classifier confidence score, $p(z_i|s_i)$, is fit using logistic regression.

4.2. Object Instance Detection and Segmentation

COCO Minival: For instance segmentation, we use 'minival2014' subset of the COCO dataset (Lin et al., 2014). This subset contains 5k images spanning over 80 categories. We report the standard COCO metric: Average Precision (averaged over all IoU thresholds).

To systematically analyze the impact of evaluation on noise and vetting, we focus evaluation efforts on the high quality test set, but simulate noisy annotations by replacing actual instance segmentation masks by their tight-fitting bounding box (the unvetted "noisy" set). We then simulate active testing where certain instances are vetted, meaning the bounding-box is replaced by the true segmentation mask.

Detection Systems: We did not implement instance segmentation algorithms ourselves, but instead utilized three sets of detection mask results produced by the authors of Mask R-CNN (He et al., 2017). These were produced by variants of the instance segmentation systems proposed in (Xie et al., 2017; Lin et al., 2017; He et al., 2017).

Learned Estimators: To compute the probability whether a detection will pass the IoU threshold with a bounding box unvetted ground-truth instance ($p(z_i|\mathcal{O})$ in Eq. 9), we train a χ^2-SVM using the vetted portion of the database. The features for an example includes the category id, the 'noisy' IoU estimate, the size of the bounding box containing the detection mask and the size of ground-truth bounding box. The training label is true whether the true IoU estimate, computed using the vetted ground-truth mask and the detection masks, is above a certain input IoU threshold.

4.3. Efficiency of active testing estimates

We measure the estimation accuracy of different combination of vetting strategies and estimators at different amount of vetting efforts. We compute the absolute error between the estimated metric and the true (fully vetted) metric and average over all classes. Averaging the absolute estimation error across classes prevents over-estimation for one class canceling out under-estimation from another class. We plot the mean and the standard deviation over 50 simulation runs of each active testing approach.

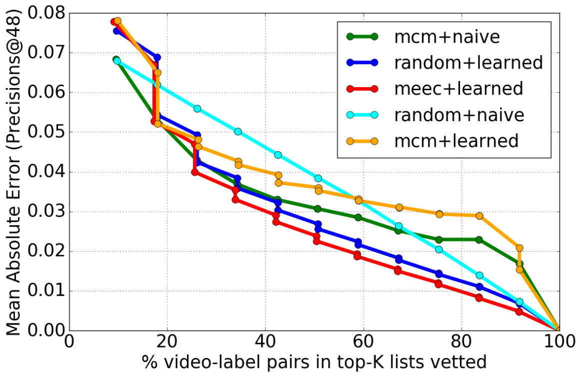


Figure 6. Decoupling the effect of model change and vetting effort for NUS-WIDE. This figure shows the reduction in estimation errors. The vertical drop at the same % vetted point indicates the reduction due to estimator quality. The slope between adjacent points indicates value of vetting examples. A steeper slope means the strategy is able to obtain a better set. In some sense, traditional active learning is concerned primarily with the vertical drop (i.e. a better model/predictor), while active testing also takes direct advantage of the slope (i.e. more vetted labels).

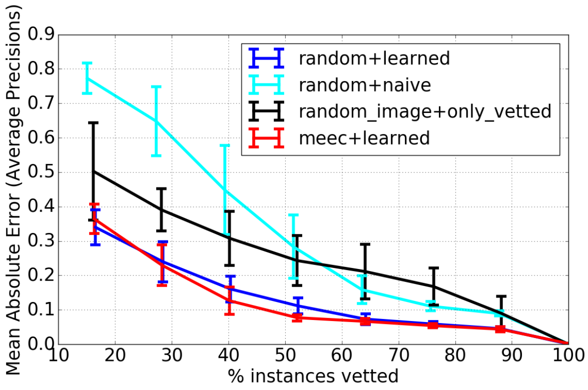


Figure 7. Results for instance segmentation. With 50% of instances vetted, our best model's estimation is 1% AP off from the true values with the standard deviation $\leq$ 1%. A smart estimator with a smarter querying strategy can make the approach more robust and efficient. Our approach has better approximation and is less prone to sample bias compared to the standard approach("random image"+ "only vetted").

Performance estimation: Figure 5 shows the results for estimating $Prec@48$ for NUSWIDE and Microvideos. The x-axis indicates the percentage of the top-k lists that are vetted. For the $Prec@K$ metric, it is only necessary to vet 100% of the top-k lists rather than 100% of the whole test set[2]. A 'random' strategy with a 'naive' estimator follows a

[2] The "vetted only" estimator is not applicable in this domain until at least K examples in each short list have been vetted and hence doesn't appear in the plots.

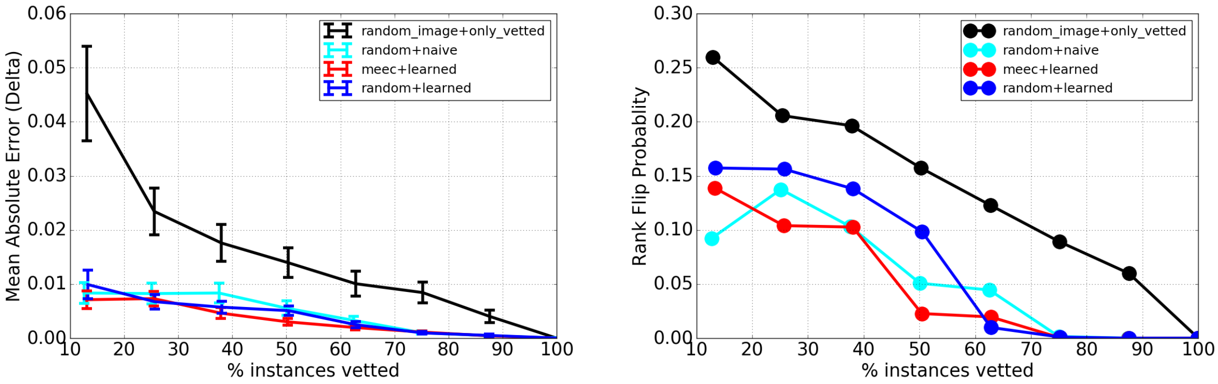


Figure 8. Relative performance differences and their relative ranking for multiple input systems. The left plot shows the mean squared errors between the current difference to the true difference. The right plot shows how often the ranking orders between two input algorithms are flipped. Both figures suggest that our active testing framework is a more robust and efficient approach toward comparing models. With 50% of the data vetted, standard approaches that evaluate on only vetted data (black curve) incorrectly rank algorithms 16% of the time, while our learned estimators with active vetting (red curve) reduce this error to 3% of the time.

linear trend since each batch of vetted examples contributes on average the same reduction in estimation error. The most confident mistake (mcm) heuristic works very well for Microvideos due to the substantial amount of undertagging. However, in more reasonable balanced settings such as NUS-WIDE, this heuristic does not perform as well. The MCM vetting strategy does not pair well with a learned estimator due to its biased sampling which quickly results in priors that overestimate the number missing tags. In contrast, the random and active *MEEC* vetting strategies offer good samples for learning a good estimator. At 50% vetting effort, *MEEC* sampling with a learned estimator on average can achieve within 2-3% of the real estimates.

Figure 6 highlights the relative value of establishing the true vetted label versus the value of vetted data in updating the estimator. In some sense, traditional active learning is concerned primarily with the vertical drop (i.e. a better model/estimator), while active testing also takes direct advantage of the slope (i.e. more vetted labels). The initial learned estimates have larger error due to small sample size, but the fitting during the first few vetting batches rapidly improves the estimator quality. Past 40% vetting effort, the estimator model parameters stabilize and remaining vetting serves to correct labels whose true value can't be predicted given the low-complexity of the estimator.

Figure 7 shows similar results for estimating the mAP for instance segmentation on COCO. The current 'gold standard' approach of estimating performance based only on the vetted subset of images leads to large errors in estimation accuracy and high variance from from small sample sizes. In the active testing framework, input algorithms are tested using the whole dataset (vetted and unvetted). Naive estimation is noticeably more accurate than vetted only and the learned estimator with uncertainty sampling further reduces both the absolute error and the variance.

Model ranking: The benefits of active testing are highlighted further when we consider the problem of ranking system performance. We are often interested not in the absolute performance number, but rather in the performance gap between different systems. We find that active testing is also valuable in this setting. Figure 8 shows the error in estimating the *performance gap* between two different instance segmentation systems as a function of the amount data vetted. This follows a similar trend as the single model performance estimation plot. Importantly, it highlights that only evaluating vetted data, though unbiased, typically produces a large error in in performance gap between models to high variance in the estimate of each individual models performance. In particular, if we use these estimates to rank two models, we will often make errors in model ranking even when relatively large amounts of the data have been vetted. Using stronger estimators, actively guided by *MEEC* sampling provide accurate rankings with substantially less vetting effort. With 50% of the data vetted, standard approaches that evaluate on only vetted data (black curve) incorrectly rank algorithms 15% of the time, while our learned estimators with active vetting (red curve) reduce this error to 3% of the time.

Conclusions We have introduced a general framework for active testing that minimizes human vetting effort by actively selecting test examples to label and using performance estimators that adapt to the statistics of the test data and the systems under test. Simple implementations of this concept demonstrate the potential for radically decreasing the human labeling effort needed to evaluate system performance for standard computer vision tasks. We anticipate this will have substantial practical value in the ongoing construction of such benchmarks.

Acknowledgement We thank Piotr Dollar and Ross Girshick for providing the instance segmentation results for the COCO dataset. This project was supported in part by NSF grants IIS-1618806 and IIS-1253538.

References

Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., and Roth, D. Generalization bounds for the area under the roc curve. *Journal of Machine Learning Research*, 2005.

Anirudh, R. and Turaga, P. Interactively test driving an object detector: Estimating performance on unlabeled data. In *WACV*, 2014.

Aslam, J. A., Pavlu, V., and Yilmaz, E. A statistical method for system evaluation using incomplete judgments. In *ACM SIGIR*. ACM, 2006.

Balcan, M.-F. and Urner, R. Active learning–modern learning theory. *Encyclopedia of Algorithms*, pp. 8–13, 2016.

Boyd, S., Cortes, C., Mohri, M., and Radovanovic, A. Accuracy at the top. In *Advances in neural information processing systems*, 2012.

Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P., and Belongie, S. Visual recognition with humans in the loop. In *ECCV*, 2010.

Chua, T.-S., Tang, J., Hong, R., Li, H., Luo, Z., and Zheng, Y. Nus-wide: a real-world web image database from national university of singapore. In *ACM international conference on image and video retrieval*, 2009.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

Dollar, P., Wojek, C., Schiele, B., and Perona, P. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 2012.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88 (2):303–338, 2010.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015.

Gao, N., Webber, W., and Oard, D. W. Reducing reliance on relevance judgments for system comparison by using expectation-maximization. In *European Conference on Information Retrieval*. Springer, 2014.

Gong, Y., Jia, Y., Leung, T., Toshev, A., and Ioffe, S. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*, 2013.

Guillaumin, M., Mensink, T., Verbeek, J., and Schmid, C. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Computer Vision, International Conference on*, 2009.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *ICCV*, 2017.

Hill, S. I., Zaragoza, H., Herbrich, R., and Rayner, P. J. Average precision and the problem of generalisation. In *ACM SIGIR Workshop on Mathematical and Formal Methods in Information Retrieval*, 2002.

Hoiem, D., Chodpathumwan, Y., and Dai, Q. Diagnosing error in object detectors. In *ECCV*, 2012.

Izadinia, H., Russell, B. C., Farhadi, A., Hoffman, M. D., and Hertzmann, A. Deep classifiers from image tags in the wild. In *Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions*, 2015.

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

Joulin, A., van der Maaten, L., Jabri, A., and Vasilache, N. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pp. 67–84. Springer, 2016.

Kamar, E., Hacker, S., and Horvitz, E. Combining human and machine intelligence in large-scale crowdsourcing. In *11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 2012.

Khattak, F. K. and Salleb-Aouissi, A. Quality control of crowd labeling through expert evaluation. In *NIPS 2nd Workshop on Computational Social Science and the Wisdom of Crowds*, 2011.

Li, D. and Kanoulas, E. Active sampling for large-scale information retrieval evaluation. In *ACM on Conference on Information and Knowledge Management*. ACM, 2017.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, J. Learning from noisy labels with distillation. *arXiv preprint arXiv:1703.02391*, 2017.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *CVPR*, 2017.

Mathias, M., Benenson, R., Pedersoli, M., and Van Gool, L. Face detection without bells and whistles. In *ECCV*. Springer, 2014.

Moffat, A., Webber, W., and Zobel, J. Strategic system comparisons via targeted relevance judgments. In *ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007.

Nguyen, P. X., Rogez, G., Fowlkes, C., and Ramanan, D. The open world of micro-videos. *arXiv preprint arXiv:1603.09439*, 2016.

Ponce, J., Berg, T. L., Everingham, M., Forsyth, D. A., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, B. C., Torralba, A., et al. Dataset issues in object recognition. In *Toward category-level object recognition*, pp. 29–48. 2006.

Rahman, M. M., Kutlu, M., Elsayed, T., and Lease, M. Efficient test collection construction via active learning. *arXiv preprint arXiv:1801.05605*, 2018.

Rashtchian, C., Young, P., Hodosh, M., and Hockenmaier, J. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 139–147. Association for Computational Linguistics, 2010.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015.

Sabharwal, A. and Sedghi, H. How good are my predictions? efficiently approximating precision-recall curves for massive datasets. In *UAI*, 2017.

Settles, B. Active learning literature survey. *University of Wisconsin, Madison*, pp. 11, 2010.

Sheshadri, A. and Lease, M. Square: A benchmark for research on computing crowd consensus. In *AAAI Conference on Human Computation and Crowdsourcing*, 2013.

Torralba, A. and Efros, A. A. Unbiased look at dataset bias. In *CVPR*, 2011.

Vapnik, V. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pp. 831–838, 1992.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., and Belongie, S. Learning from noisy large-scale datasets with minimal supervision. *arXiv preprint arXiv:1701.01619*, 2017.

Vijayanarasimhan, S. and Grauman, K. Large-scale live active learning: Training object detectors with crawled data and crowds. *IJCV*, 2014.

Wah, C., Branson, S., Perona, P., and Belongie, S. Multiclass recognition and part localization with humans in the loop. In *ICCV*, 2011.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Van Gool, L. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

Welinder, P., Welling, M., and Perona, P. A lazy man's approach to benchmarking: Semisupervised classifier evaluation and recalibration. In *CVPR*, 2013.

Wu, B., Lyu, S., and Ghanem, B. Ml-mg: multi-label learning with missing labels using a mixed graph. In *ICCV*, 2015.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

Yilmaz, E. and Aslam, J. A. Estimating average precision with incomplete and imperfect judgments. In *ACM international conference on Information and knowledge management*. ACM, 2006.

Yu, H.-F., Jain, P., Kar, P., and Dhillon, I. S. Large-scale multi-label learning with missing labels. In *ICML*, 2014.

Active Testing: Supplementary Materials

Phuc Nguyen [1] **Deva Ramanan** [2] **Charless Fowlkes** [1]

1. Expected Precision and Average Precision

Let Precision and Recall at K be defined as

$$P_k = \frac{1}{k}\sum_{i\le k} z_i \tag{1}$$

and,

$$R_k = \frac{1}{N_p}\sum_{i\le k} z_i \tag{2}$$

where N_p is the number of positive instances in the whole set. The average precision is given by integrating precision with respect to recall:

$$\begin{aligned} AP &= \sum_k (R_k - R_{k-1})P_k \\ &= \sum_k \Big(\frac{1}{N_p} z_k\Big) P_k \\ &= \sum_k \Big(\frac{1}{N_p} z_k\Big)\Big(\frac{1}{k}\sum_{i\le k} z_i\Big) \\ &= \frac{1}{N_p}\sum_k \frac{z_k}{k}\sum_{i\le k} z_i \end{aligned} \tag{3}$$

We would like to compute expectations when some z_i are unobserved. For notational convenience, let $p_i = P(z_i = 1|\mathcal{O})$ when z_i is unobserved and $\tilde{z}_i$ be the observed value when the ground-truth associated with example i is vetted. We can then compute expected $Prec@K$ as:

$$\begin{aligned} E[Prec@K] &= \frac{1}{K}\sum_{i\le K} E[z_i] \\ &= \frac{1}{K}\left(\sum_{i\le K, i\in V} \tilde{z}_i + \sum_{i\le K, i\in U} p_i\right) \end{aligned} \tag{4}$$

And the expected change for this metric is given by:

$$\begin{aligned} E_{p(z_i|V)}\left[\Delta_i(z_i)\right] &= p_i\frac{1}{K}|1-p_i| + (1-p_i)\frac{1}{K}|0-p_i| \\ &= \frac{2}{K}p_i(1-p_i) \end{aligned} \tag{5}$$

where we write $p_i = p(z_i = 1|\mathcal{O})$.

Expected AP is more interesting because it includes products of of the z_i.

$$\begin{aligned} E[AP] &= \frac{1}{N_p}\sum_k \frac{1}{k} E[z_k \sum_{i\le k} z_i] \\ &= \frac{1}{N_p}\sum_k \frac{1}{k}\sum_{i\le k} E[z_k z_i] \end{aligned}$$

We note that in our application of evaluating instance segmentation, the quantity N_p is known prior to vetting. In other settings, it may also be a random variable that depends on the vetting outcomes. In the following derivation, we temporarily drop the constant $\frac{1}{N_p}$ to reduce notational clutter.

Assuming independence of z_i and z_k, we have:

$$\begin{aligned} E[z_i] &= p_i \\ E[z_i z_k] &= p_i p_k \end{aligned}$$

Expanding the vetted and unvetted terms we can compute:

$$\begin{aligned} E[AP] =& \sum_k \frac{1}{k}\sum_{i\le k} E[z_k z_i] \\ =& \sum_{k\in V}\frac{1}{k}(\sum_{i\le k, i\in V} z_k z_i) + \sum_{k\in V}\frac{1}{k}(\sum_{i\le k, i\in U} z_k p_i) \\ &+ \sum_{k\in U}\frac{1}{k}(\sum_{i\le k, i\in V} p_k z_i) + \sum_{k\in U}\frac{1}{k}(\sum_{i\le k, i\in U} p_k p_i) \end{aligned}$$

[1] University of California, Irvine [2] Carnegie Mellon University. Correspondence to: Phuc Nguyen <nguyenpx@uci.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

which can be written a bit more compactly as:

$$E[AP] = \Big(\sum_{k\in V}(\frac{z_k}{k}E[Prec@k]) + \sum_{k\in U}(\frac{p_k}{k}E[Prec@k])\Big)$$

We would like to compute the change in $E[AP]$ when we vet some example. Before vetting sample j, we have:

$$\begin{aligned}E[AP] = &\sum_{k\in V}\frac{1}{k}\left[\sum_{i\leq k, i\in V} z_k z_i\right] \\ &+ \sum_{k\in V}\frac{1}{k}\left[\sum_{i\leq k, i\in U\setminus j} z_k p_i + z_k p_j \delta[j\leq k]\right] \\ &+ \sum_{k\in U\setminus j}\frac{1}{k}\left[\sum_{i\leq k, i\in V} p_k z_i\right] + \frac{1}{j}\left[\sum_{i\leq j, i\in V} p_j z_i\right] \\ &+ \sum_{k\in U\setminus j}\frac{1}{k}\left[\sum_{i\leq k, i\in U\setminus j} p_k p_i + p_k p_j \delta[j\leq k]\right] \\ &+ \frac{1}{j}\left[\sum_{i\in U\setminus j, i\leq j} p_j p_i + p_j p_j\right]\end{aligned}$$

After vetting the example j, we have:

$$\begin{aligned}E[AP|z_j] = &\sum_{k\in V}\frac{1}{k}\sum_{i\leq k, i\in V} z_k z_i \\ &+ \sum_{k\in V}\frac{1}{k}\left[\sum_{i\leq k, i\in U\setminus j} z_k p_i + z_k z_j \delta[j\leq k]\right] \\ &+ \sum_{k\in U\setminus j}\frac{1}{k}\left[\sum_{i\leq k, i\in V} p_k z_i\right] + \frac{1}{j}\left[\sum_{i\leq j, i\in V} z_j z_i\right] \\ &+ \sum_{k\in U\setminus j}\frac{1}{k}\left[\sum_{i\leq k, i\in U\setminus j} p_k p_i + p_k z_j \delta[j\leq k]\right] \\ &+ \frac{1}{j}\left[\sum_{i\leq j, i\in U\setminus j} z_j p_i + z_j z_j\right]\end{aligned}$$

The difference between these estimates is,

$$\begin{aligned}\Delta(z_j) &= E[AP|z_j] - E[AP] \\ &= \sum_{k\in V}\frac{1}{k}z_k(z_j - p_j)\delta[j\leq k] + \\ &\quad \frac{1}{j}\sum_{i\leq j, i\in V}(z_j - p_j)z_i + \\ &\quad \sum_{k\in U\setminus j}\frac{1}{k}\left[p_k(z_j - p_j)\delta[j\leq k]\right] + \\ &\quad \frac{1}{j}\left[\sum_{i\leq j, i\in U\setminus j}(z_j p_i + z_j z_j - p_j p_i - p_j p_j)\right]\end{aligned}$$

The expected reduction given our estimator for z_j is

$$E[\Delta] = p_j\Delta(z_j = 1) + (1 - p_j)\Delta(z_j = 0)$$

where:

$$\begin{aligned}\Delta(z_j = 0) = &\sum_{k\in V}\frac{1}{k}z_k(-p_j)\delta[j\leq k] + \\ &\frac{1}{j}\sum_{i\leq j, i\in V}(-p_j)z_i + \\ &\sum_{k\in U\setminus j}\frac{1}{k}p_k(-p_j)\delta[j\leq k] + \\ &\frac{1}{j}\left[\sum_{i\leq j, i\in U\setminus j}(-p_j p_i - p_j p_j)\right]\end{aligned}$$

$$\begin{aligned}\Delta(z_j = 1) = &\sum_{k\in V}\frac{1}{k}z_k(1 - p_j)\delta[j\leq k] + \\ &\frac{1}{j}\sum_{i\leq j, i\in V}(1 - p_j)z_i + \\ &\sum_{k\in U\setminus j}\frac{1}{k}p_k(1 - p_j)\delta[j\leq k] + \\ &\frac{1}{j}\left[\sum_{i\leq j, i\in U\setminus j}(p_i + 1 - p_j p_i - p_j p_j)\right]\end{aligned}$$

Let's just look at the first pair of corresponding terms of $E[\Delta]$,

$$(1-p_j)\sum_{k\in V}\frac{1}{k}z_k(-p_j)\delta[j\leq k] + p_j\sum_{k\in V}\frac{1}{k}z_k(1-p_j)\delta[j\leq k]$$

It is clear to see that the above summation equals 0. This is also true for the second and third terms,

$$(1-p_j)\frac{1}{j}\sum_{i\leq j, i\in V}(-p_j)z_i + p_j\frac{1}{j}\sum_{i\leq j, i\in V}(1-p_j)z_i$$

$$(1-p_j)\sum_{k\in U\setminus j}\frac{1}{k}p_k(-p_j)\delta[j\leq k]+p_j\sum_{k\in U\setminus j}\frac{1}{k}p_k(1-p_j)\delta[j\leq k]$$

Only the last pair of terms remains, and simplifies as:

$$\begin{aligned}
E[\Delta] &= -\frac{1}{j}\sum_{i\leq j, i\in U\setminus j} p_j^3 - p_j + p_j^2(1-p_j) \\
&= -\frac{1}{j}\sum_{i\leq j, i\in U\setminus j} p_j^3 - p_j + p_j^2 - p_j^3 \\
&= -\frac{1}{j}\sum_{i\leq j, i\in U\setminus j} -p_j + p_j^2 \\
&= \frac{1}{j}\sum_{i\leq j, i\in U\setminus j} p_j(1-p_j)
\end{aligned}$$

Let r_j be the proportion of unvetted examples scoring higher than example j:

$$\begin{aligned}
r_j &= \frac{|\{i\in U : i\leq j\}|}{|\{i\in U\cup V : i\leq j\}|} \\
&= \frac{1}{j}\sum_{i<j}\delta(i\in U)
\end{aligned}$$

Putting back in the constant scaling yields the expression given in the paper:

$$\begin{aligned}
E[\Delta] &= \frac{1}{N_p}\frac{1}{j}\sum_{i\leq j, i\in U\setminus j} p_j(1-p_j) \\
&= \frac{1}{N_p} r_j p_j(1-p_j)
\end{aligned}$$

The term is largest when p_j is 0.5 and decrease as it approaches 0 or 1. The term also decreases when there are many unvetted examples that score higher than j since they have relatively more impact on the AP.

2. Estimator for multilabel classification

Here we derive the basis for Equation 4 in the main paper.

$$\begin{aligned}
p(z_i|y_i,s_i) &= \frac{p(z_i,y_i,s_i)}{\sum_{v\in\{0,1\}}p(z_i=v,y_i,s_i)} \\
&= \frac{p(y_i|z_i,s_i)p(z_i|s_i)}{\sum_{v\in\{0,1\}}p(y_i|z_i,s_i)p(z_i|s_i)}
\end{aligned}$$

We assume that given the true label, z_i, the observed label y_i is conditionally independent of the classifier score, s_i. With $p(y_i|z_i,s_i)=p(y_i|z_i)$, the expression simplifies to,

$$p(z_i|y_i,s_i) = \frac{p(y_i|z_i)p(z_i|s_i)}{\sum_{v\in\{0,1\}}p(y_i|z_i)p(z_i|s_i)}$$

3. Additional Results

Figure 1 shows the results of estimating absolute precision@48 for the multilabel classification tasks on both NUS-WIDE and Microvideos datasets. In contrast, plots in the main paper show the total absolute error from the true value.

References

Boyd, S., Cortes, C., Mohri, M., and Radovanovic, A. Accuracy at the top. In *Advances in neural information processing systems*, 2012.

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

Vapnik, V. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pp. 831–838, 1992.

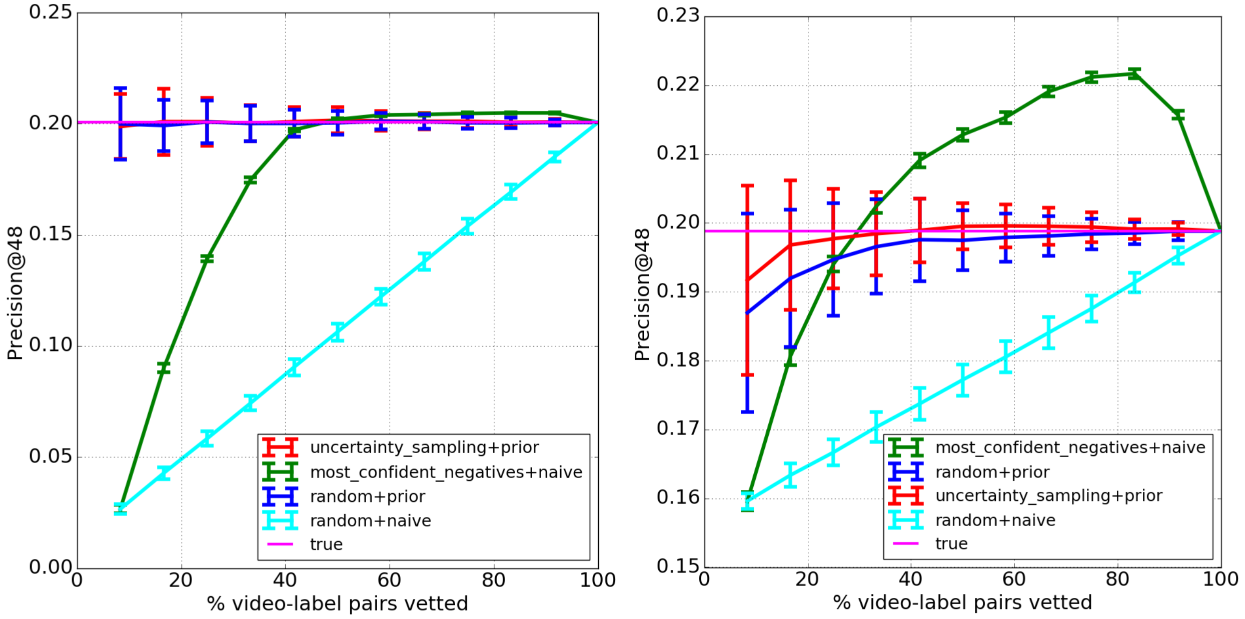


Figure 1. Results for multi-label classification task. The figures show the mean and standard deviation of the estimated Precision@K at different amount of annotation efforts. Using a fairly simple estimator and vetting strategy, the proposed framework can estimate the performance very closely to the true values.