To appear, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June, 2010.

Far-Sighted Active Learning on a Budget for Image and Video Recognition

Sudheendra Vijayanarasimhan, Prateek Jain, and Kristen Grauman
University of Texas at Austin
{svnaras,pjain,grauman}@cs.utexas.edu

Abstract

Active learning methods aim to select the most informative unlabeled instances to label first, and can help to focus image or video annotations on the examples that will most improve a recognition system. However, most existing methods only make myopic queries for a single label at a time, retraining at each iteration. We consider the problem where at each iteration the active learner must select a set *of examples meeting a given* budget *of supervision, where the budget is determined by the funds (or time) available to spend on annotation. We formulate the budgeted selection task as a continuous optimization problem where we determine which subset of possible queries should maximize the improvement to the classifier's objective, without overspending the budget. To ensure far-sighted batch requests, we show how to incorporate the predicted change in the model that the candidate examples will induce. We demonstrate the proposed algorithm on three datasets for object recognition, activity recognition, and content-based retrieval, and we show its clear practical advantages over random, myopic, and batch selection baselines.*

1. Introduction

The accuracy of a supervised classifier is often strongly linked to the quantity of the annotated training data available—having access to more examples means each category's variability can be more easily captured. However, not all examples are equally informative, and an arbitrary unlabeled example may even be redundant. *Active learning* methods provide a way to automatically pinpoint informative examples for which labels should be requested, thereby reducing supervision without sacrificing accuracy in the model [1, 2, 3, 4, 5]. Recent results have shown that active selection can benefit image and video recognition systems, and save human annotators from excessive image labeling, segmentation, relevance feedback, or other data collection tasks [6, 7, 8, 9, 10, 11, 12].

However, there are two limiting assumptions often made in active learning. First, most methods select a *single* unlabeled instance to query at a time, retraining the classifier at each iteration once the label is obtained. This is a problem since retraining after every single label is expensive, and it is often preferable to farm out a *batch* of good queries at once. Systems such as Mechanical Turk or LabelMe provide access to multiple distributed annotators simultaneously, but an active learning system that needs to repeatedly go offline and compute the next annotation request cannot take advantage of such resources. Second, existing techniques typically assume that all examples require the same amount of manual effort to label, and thus aim to minimize the total number of queries made. In reality, the annotation cost associated with labeling different examples often varies, sometimes significantly. For instance, the longer the video, the longer it will take to watch it and annotate its contents; the more sophisticated the image query, the more we may need to pay a human labeler to answer it.

The technical problem of selecting a good set of examples at once is challenging, since one must take care to avoid overlapping information; i.e., it is wasteful to ask a batch of similar questions. Furthermore, it is risky to formulate a large selection based only on the current model's view of the data: some examples within large sets may lead to significant changes to the classifier that ultimately invalidate the perceived value of others that were selected. While a few "batch-mode" active learning strategies have been proposed in the machine learning literature [13, 14, 12, 15], none consider how to balance the joint selection with annotation costs. Meanwhile, current active selection approaches that do account for labeling cost lead to a myopic selection of a single request at a time [4, 16, 17, 18, 11].

In this paper, we formalize the problem of far-sighted active learning on a budget. At each iteration the active learner is allowed to choose a set of examples to get labeled, provided the total sum of annotation costs associated with the selected examples is under a given budget. We propose a novel method for optimally selecting a set of examples for a support vector machine (SVM) classifier under these conditions. Given an unlabeled pool of data where each example has an associated annotation cost, we introduce a set of instance selection variables. We formulate an optimization problem to learn the maximum margin hyperplane along with the instance variables that minimize the empirical risk (on both the labeled data and selected unlabeled points), while satisfying the given budget constraint. We then relax it to a continuous optimization problem that can be decomposed into two strictly convex optimization problems loosely coupled in the hyperplane parameters and selection variables. We devise a monotonically convergent alternating minimization algorithm to compute the solution.

The proposed approach is the first batch selection strategy that is sensitive to the costs of labeling, and the first method to allow sets of training examples to be chosen so as to meet a prescribed budget. The efficiency of the component optimization steps makes it rather scalable to large unlabeled data pools. Furthermore, in contrast to previous methods, our approach considers how much the classifier objective changes if we were to obtain the most probable labels on the candidate examples for selection. We find that this aspect is critical to performance, particularly in the scenario where one wants to set a large budget.

We validate our method on benchmark datasets for three applications: object recognition, activity recognition, and image retrieval. We demonstrate the advantages of our approach compared to passive, myopic, and batch selection baselines, and show its effectiveness across a range of budgets. Our results indicate that budgeted selection is crucial for efficient active learning in practical scenarios, clearly outperforming conventional myopic selection techniques.

2. Background and Related Work

Various active learning strategies have been proposed to choose the best example to query for a label, including methods based on uncertainty sampling [5], entropy [1], reducing the version space [2], or predicting reductions in risk [3, 4]. See [19] for a comprehensive survey.

Vision researchers have explored active strategies to annotation in order to learn more efficiently about object categories [7, 8, 9, 10, 11], or to interactively request relevance feedback from a user [6]. However, in contrast to this work, the previous approaches focus on myopic selection.

Traditionally, active methods have assumed that each training example requires the same amount of effort (cost) to label. The danger in doing so is that this may favor choosing expensive labels first, leading to inflated portrayals of accuracy gains when measured relative to the number of queries [16]. Recent work suggests ways to instead balance the informativeness criteria against the expected labeling costs for various learning problems [4, 16, 20, 21, 18],including object recognition [18, 11].

The authors of [17] explore a form of budgeted learning where one can interactively accumulate answers about a particular *test* example, repeatedly deciding if a further query is worthwhile before making a prediction. Whereas their approach builds "active classifiers", the form of budgeted learning we consider pertains to the selection process among unlabeled candidate *training* examples.

When one has access to multiple "labelers" at once (e.g., on MTurk [22, 23]), a batch selection would be more effective. A few batch-mode active learning methods have been proposed [14, 24, 15], including one for vision [12]. Batch selection calls for more than a selection of the N-best queries at a given iteration, since such a greedy strategy does not account for possible overlap in information. Instead, selection functions try to balance informativeness with the so-called *diversity* among the selected set [14, 12, 24].

Unfortunately, by relying on the current classifier to estimate uncertainty, these functions' performance can degrade with very large batches; balancing uncertainty and diversity properly can also require good heuristics. In contrast, our solution makes selections that account for model changes that may result from the not-yet-labeled points. The method of [15] also integrates model changes and formulates selection as an optimization problem. However, our solution limits the objective to include selected examples rather than the entire unlabeled set as in [15], which may limit the influence of erroneous "optimistic" labels. Our method differs foremost, however, in its ability to work on a budget. Unlike any previous work, our approach enables variably-sized batches to be chosen in response to real cost estimates; our experiments demonstrate the practical advantages.

3. Approach: Budgeted Batch Selection

Given a preliminary recognition model and a budget for annotations to improve the training set, our method considers all the available unlabeled image data and computes the set of recommended requests that are jointly most informative and fall within the budget.

Below, we first formally define the problem of budgeted selection, and overview the main idea of our approach. In Sec. 3.2 we present the detailed formulation and algorithm.

3.1. Problem Definition and Overview

We consider the problem of actively selecting a batch of examples to label, where the contents of the batch must be constrained by some budget. Formally, let $L = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$ denote a set of l initially labeled examples, where $y_i \in \{+1, -1\}$. Let $U = \{x_{l+1}, x_{l+2}, \ldots, x_{l+u}\}$ denote an unlabeled pool from which examples can be selected and given to an oracle labeler. Each unlabeled example x_i is associated with a labeling cost c_i, which measures the manual effort required to obtain a label for x_i. Note that the cost varies per example.

At each iteration, a set of examples $S =$

$\{x_{k_1}, x_{k_2}, \ldots, x_{k_n}\} \subseteq U$ can be selected for labeling, as long as the total annotation cost of the selection does not exceed a specified budget T. That is, $\sum_{j=1}^{n} c_{k_j} \leq T$. Since the costs vary, the number of selected examples n is not fixed. The goal is therefore to maximally utilize the given budget T by selecting the set S that is expected to produce the most gain in the classifier's performance. After obtaining labels for the chosen set, the classifier will be retrained, and the process can repeat, one batch at a time.

A naive approach to this problem, which we refer to as *Myopic Active Batch* (MAB) learning, would be to greedily choose the top most uncertain examples according to the current classifier that fit under the given budget—in other words, to rank the points in descending order by their uncertainty, and start adding them to the set S until the total budget is exhausted. However, such an approach ignores the information overlap between the selected examples.

Existing methods counter this problem by choosing a set that contains both examples that are uncertain and that are mutually diverse [14, 12]. Aside from needing good heuristics to balance the two properties, estimating uncertainty based on the *current classifier* (e.g., using the distance from the margin for an SVM [2]) also fails to capture how uncertainty will change once the selected examples are added to the labeled set and the model's parameters are retrained. For large batches of examples this can be especially problematic. In addition, existing methods are specifically targeted at choosing a fixed number of examples at each iteration, but a variable-sized batch may more optimally use labeling resources (i.e., a fixed-size batch must take n total examples, whereas a more effective selection might entail choosing a couple of the more expensive examples together with a set of $<< n$ cheaper ones).

Therefore, we propose an approach that directly targets the amount of reduction in the SVM objective that is to be expected by choosing a given set of examples under a budget. We call this *budgeted batch selection*. The main idea is as follows: we introduce an indicator variable over the unlabeled examples, and formulate a continuous optimization problem to determine which subset of possible queries should maximize the improvement to the classifier's objective, without overspending the budget. When fixing the selection variables, the optimization reduces to that of a standard SVM objective function, which can be solved efficiently; when fixing the model parameters, the selection variables are computed via linear programming. Because we incorporate the predicted change in the model that the candidate examples will induce, the method is "far-sighted" in terms of the effects of the entire batch.

3.2. Formulation and Algorithm

Given a set of labeled examples L, the SVM objective seeks the optimal separating hyperplane defined by parameters (w, b):

$$\begin{aligned} \arg\min_{w,b,\epsilon} \quad & \frac{1}{2}||w||^2 + C \sum_{(x_i,y_i)\in L} \epsilon_i, \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \epsilon_i,\ (x_i, y_i) \in L, \\ & \epsilon_i \geq 0, \end{aligned} \tag{1}$$

where each ϵ_i denotes the hinge loss on x_i, and C denotes the constant regularization penalty. This familiar SVM objective simultaneously minimizes the classification error on the training examples while maximizing the margin of separation between the positives and negatives.

Let A and B be two (possibly distinct) sets of labeled examples. To aid in notation below, we define an intermediate cost function, which takes parameters f_A and B:

$$g(f_A, B) = \frac{1}{2}||w_A||^2 + C\hat{R}_B^A, \tag{2}$$

where f_A denotes the SVM hyperplane parameters $f_A = (w_A, b_A)$ obtained by training on set A, and $\hat{R}_B^A = \sum_{(x_i,y_i)\in B} \epsilon_i^A$ denotes the empirical loss incurred by model f_A over the set B, and

$$\epsilon_i^A = \max(0, 1 - y_i(w_A^T x_i + b_A)) \tag{3}$$

denotes the hinge loss on x_i resulting from the model f_A. Note that the cost measured by $g(f_A, B)$ evaluates a margin term $\frac{1}{2}||w_A||^2$ (which reflects generalization ability) using the solution according to labeled data A, whereas it evaluates losses (which reflect misclassifications) on examples in B using the model f_A.

We want both the labels on the candidate selection sets as well as the existing labeled data to simultaneously influence the batch selection. As the points in a candidate set S are as yet unlabeled, we can only estimate the most "optimistic" cost reduction by maximizing over all possible labels on S. In the following, we use the term *optimistic labels* (borrowed from [15]) to refer to a label assignment for unlabeled points under which cost is maximally reduced.

Let $Y^* = \{y_{k_1}, \ldots, y_{k_n}\}$, be the set of optimistic labels associated with the examples in the optimal selection $S^* \subseteq U$, where $Y^* \in \{+1, -1\}^n$, for $n = |S^*|$. We want to select (S^*, Y^*) such that together they yield the maximal cost reduction, as measured by the cost produced *before* their addition to the labeled set versus the cost produced *after* they are added. Specifically, we want:

$$\begin{aligned} (S^*, Y^*) = \arg\min_{S\subseteq U, Y} \quad & g(f_{L'}, L') - g\left(f_L,\ L \cup (S, Y_L)\right), \\ \text{s.t.} \quad & \sum_{x_i \in S} c_i \leq T, \end{aligned} \tag{4}$$

where $L' = L \cup (S, Y)$—that is, the labeled set expanded with some label assignment on S—and Y_L denotes the labels obtained by classifying S using f_L. The last inequality reflects the budget constraint limiting total annotation cost among selected examples to T. Note that the first term in the above objective measures the classification error on $L \cup (S, Y)$ and the margin when training using both L and (S, Y), while the second term measures both the margin and the classification error for the selected examples under the "old" model f_L, which is trained only on L. Thus, the optimal (S^*, Y^*) results in the maximal reduction in the SVM objective when considering optimistic labels [1].

To solve this optimization problem, we first expand the representation of the unlabeled set so that each unlabeled example appears as two examples labeled with both possible classes. Formally, we expand U to also include:

$$\begin{aligned} x_i &= x_{i-u}, \text{for } i \in [l+u+1, \dots, l+2u], \\ y_i &= +1, \quad \text{for } i \in [l+1, \dots, l+u], \\ y_i &= -1, \quad \text{for } i \in [l+u+1, \dots, l+2u]. \end{aligned} \tag{5}$$

From here on, U represents the expanded unlabeled set. We then introduce a vector of indicator variables $q \in [0,1]^{2u}$, where $q_j = 1$ denotes that example $x_{l+j} \in S$, and $q_j = 0$ denotes that it is not. Let Y_U denote the set of labels on all unlabeled examples, which includes the labels Y for selection S. Now redefining $L' = L \cup (S, Y_U)$ we can rewrite the first g term from Eqn. 4 as:

$$\begin{aligned} g(f_{L'}, L') &= \frac{1}{2}||w_{L'}||^2 + C\hat{R}_{L'}^{L'} \\ &= \frac{1}{2}||w_{L'}||^2 + C\hat{R}_{L}^{L'} + C_u \hat{R}_{(S,Y_U)}^{L'}, \\ &= \frac{1}{2}||w_{L'}||^2 + C\hat{R}_{L}^{L'} + C_u \sum_{j=1}^{2u} q_j \epsilon_{l+j}^{L'}, \end{aligned} \tag{6}$$

where C_u is a constant regularization penalty for the selected unlabeled examples. Here $f_{L'}$ is obtained by solving the optimization problem in Eqn. 1 with the set $L \cup (S, Y)$, and the values for each ϵ_i are also based on this model $f_{L'}$ (and hence the labels Y_U), as denoted by the $\epsilon_{l+j}^{L'}$ terms.

Substituting $g(f_{L'}, L')$ from Eqn. 6 into Eqn. 4, the desired selection problem can now be written as:

$$\begin{aligned} \min_{w,b,q} \ & \frac{1}{2}||w||^2 + C \sum_{x_i \in L} \epsilon_i + C_u \sum_{j=1}^{2u} \epsilon_{l+j} q_j - C_u \sum_{j=1}^{2u} \epsilon_{l+j}^{L} q_j, \\ \text{s.t. } \ & y_i(w^T x_i + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, \quad 1 \leq i \leq l+2u, \\ & \sum_{j=1}^{2u} q_j c_j \leq T, \\ & q_j + q_{u+j} \leq 1, \quad 1 \leq j \leq u, \\ & q_j \in \{0,1\}, \qquad 1 \leq j \leq 2u, \end{aligned} \tag{7}$$

[1] Note that our motivation for using optimistic labels above, rather than the expected value over all labels, is to reduce the impact of the current model's ambiguity.

where $L' = L \cup (S, Y_U)$. Note that our encoding of the indicator means that q^* itself represents (S^*, Y^*) from Eqn. 4, and similarly the expanded labeled set L' is a function of q. We drop the superscripts L' for the ϵ_i variables for clarity since L' is now a parameter that we are optimizing over. Note that the first two terms of Eqn. 6 for $g(f_L, L \cup (S, Y))$ are constant w.r.t. the optimization variables and thus are ignored. The last term reflects the loss incurred for examples in S using a model L that does *not* account for labels Y_U, whereas the middle two reflect errors after its inclusion.

Although Eqn. 7 includes a constraint for every unlabeled example $x_{l+j} \in U$, since the penalty for the corresponding slack variable ϵ_{l+j} is zero whenever q_j is zero, the constraint only affects the annotation cost for examples with non-zero q_j, that is, for $x_{l+j} \in S$. The constraint on pairs of q variables $(q_j + q_{u+j})$ reflects that only one of the labels ($+1$ or -1) can be chosen for an unlabeled example.

The optimization problem defined above is an integer programming problem which in general is NP-hard. Hence, we first relax it to a continuous optimization problem by allowing the q variables to take values between $(0,\ 1)$. Now the above objective can be seen as two different optimization problems loosely coupled by the term $C_u \sum_{j=1}^{2u} \epsilon_{l+j} q_j$: one on (w, b) and the other on q, both of which are convex. Fixing q, the minimization over w can be done by standard convex quadratic programming. Fixing w, the minimization over q is a convex linear programming problem.

To solve the relaxed problem, we devise an iterative alternating minimization procedure that is guaranteed to converge to a local optimum of the objective function. Assuming q to be constant, Eqn. 7 reduces to

$$\begin{aligned} (w^*, b^*) = \arg\min_{w,b} \ & \frac{1}{2}||w||^2 + C \sum_{(x_i, y_i) \in L} \epsilon_i + C_u \sum_{j=1}^{2u} \epsilon_{l+j} q_j, \\ \text{s.t. } \ & y_i(w^T x_i + b) \geq 1 - \epsilon_i, \epsilon_i \geq 0, \quad (x_i, y_i) \in L, \\ & y_i(w^T x_{l+j} + b) \geq 1 - \epsilon_{l+j}, \\ & \epsilon_{l+j} \geq 0, \ \ x_{l+j} \in U, \ \ y_{l+j} \in Y_U. \end{aligned} \tag{8}$$

Note that this objective has a very similar form to that of the transductive SVM, as first proposed in [25]. Importantly, unlike the transductive SVM, in this case the inclusion of the indicator vector q means we penalize labeling errors on unlabeled data in S only, which is a subset of all unlabeled examples. Moreover, for a fixed q the problem reduces to that of the standard SVM problem, where the annotation cost for the unlabeled examples is a function of the q variables. Hence, for a given q, we can efficiently optimize (w, b) and their optimistic labels for the selected batch.

Conversely, fixing the variables (w, b) and relaxing the indicator vector as $q \in (0,1)^{2u}$, Eqn. 7 reduces to

$$q^* = \arg\min_q \; C_u \sum_{j=1}^{2u} \epsilon_{l+j} q_j - C_u \sum_{j=1}^{2u} \epsilon_{l+j}^L q_j,$$
$$\text{s.t.} \; \sum_{j=1}^{2u} q_j c_j \leq T,$$
$$q_j + q_{u+j} \leq 1, \quad 1 \leq j \leq u,$$
$$0 \leq q_j \leq 1, \qquad 1 \leq j \leq 2u. \tag{9}$$

The above problem is a linear programming problem in q and can be solved using standard methods like an interior point method.[2] The ϵ_{l+j} variables depend on the current solution for (w^*, b^*) from Eqn. 8, whereas ϵ_{l+j}^L is a function of the parameters (w_L, b_L)—which are obtained by training on L alone—and Y_L, the true labels obtained so far.

Finally, by alternating between Eqns. 8 and 9, we can compute the batch selection meeting the given budget that is expected to most improve the classifier. We always initialize the ϵ values to 0, which corresponds to initializing our method with the myopic solution. We form S^* by choosing the examples with the largest q_i that fit the given budget T. Algorithm 1 provides pseudo-code for the procedure.

Note that the constraints on $\{w, b, \epsilon\}$ in Eqn.7 are independent of q. Similarly, constraints on q are independent of $\{w, b, \epsilon\}$. Hence, fixing q and optimizing for $\{w, b, \epsilon\}$ decreases the objective function (Step 5 in Alg. 1). Similarly, Step 6 also decreases the objective function. Hence, our algorithm converges monotonically. In fact, with a stronger analysis, it is easy to show that our algorithm converges to a local optimal of the objective function.

Our solution is quite efficient since it uses an LP and QP for which several efficient solvers exist. In our experiments, convergence typically occurs in $\sim$12 iterations. Our Matlab code takes about 0.6 secs per batch selection for 200 unlabeled examples, and 4 mins for 5000 examples.

3.3. Summary: Using the Budgeted Selection

Our approach can be used for active training of any SVM classification problem. The inputs are an initial training set containing some labeled examples of the categories of interest, the number of selection iterations, an unlabeled pool of data, and the available budget. In practice, one would set this budget according to the resources available—for example, the money one is willing to spend on Mechanical Turk to get a training set for the task. We construct the initial classifier, and then for each iteration, solve for the indicator vector specifying which set of unlabeled data objects should be annotated next. For unlabeled data with non-uniform annotation costs, each resulting request will consist of a variable number of items (images, video clips). Once these tasks are completed (either sequentially, or in parallel by a team of annotators), the labeled set is expanded accordingly, the classifier is updated, and budgeted selection repeats. The final output is the trained classifier.

[2]In the implementation, we need to add slack on T since with varying annotation costs per example one can only hit the budget T as closely as possible, but for clarity of presentation we omit it in the notation above.

Algorithm 1 Budgeted Batch Active Learning (BBAL)

```
Require: Labeled data - $L$, Unlabeled data - $U$,
         Current loss on unlabeled data - $\epsilon_i^L$,
         Labeling costs - $c = [c_1, \ldots, c_u]$, Budget - $T$,
         Parameters - $C$, $C_u$, $\zeta$.
 1: Initialize $\epsilon_{l+j}$ = 0, for $j = 1, \ldots, 2u$.
 2: $Y_U = \{y_1, \ldots, y_{2u}\}$, set as in Eqn. 5.
 3: $\mathcal{C}(q_{old})$ = $\infty$, where $\mathcal{C}(\cdot)$ denotes objective in Eqn. 7
 4: repeat
 5:     $q$ = solve_linear_program ($\epsilon$, $\epsilon^L$, $c$, $T$)      // Eqn. 9
 6:     $[w, b]$ = svm($L \cup (U, Y_U)$, $C \cup qC_u$)      // Eqn. 8
 7:     Compute $\epsilon$ using $Y_U$, $w$, and $b$.
 8:     $\mathcal{C}(q_{new})$ = $q$.
 9: until convergence.
    Set $q_j$ = max($q_j$, $q_{j+u}$), for $j = 1, \ldots, u$.
10: return Set $S^* = \cup_{q_j > 0} \; x_{l+j}$, for $j = 1, \ldots, u$.
```

4. Results

We demonstrate our approach with multiple visual recognition applications. The main goal of our experiments is to demonstrate the advantage of maximally utilizing budgets of any size, and to validate the importance of using the change in the classifier objective when choosing large batches. To show these things, we consider three baselines:

- **Passive selection**: randomly chooses examples to label. To implement this on a budget, we randomly draw from the unlabeled pool until the budget is exhausted.
- **Myopic active batch** learner (MAB): greedily takes the examples closest to the margin whose summed annotation costs come in under budget. This is a batch-mode extension of [2], a common approach for SVMs.
- **Batch-mode active** learner (BMAL): a state-of-the-art approach that selects batches of a fixed size [12]. Like our method, it considers an SVM objective, but it does not include the model's expected change during selection, and it ignores per-example annotation costs.

We emphasize that, to our knowledge, no existing method allows batch selection on a budget, making these the best three baselines to analyze.

Datasets and Implementation Details: We use three publicly available benchmark datasets: SIVAL for object recognition, Hollywood for activity recognition, and Corel for CBIR. The first two consist of examples that require variable effort to annotate, allowing us to study the advantages of selecting requests to meet a budget. The third al-

Dataset	Annotation Cost (secs)				
	min	max	mean	median	total
SIVAL	4	202	31.9	32	6752
Hollywood	1.64	92.7	15.4	8.7	2476.7

Table 1. Distribution of annotation costs on SIVAL and HOHA.

lows us to make direct comparisons with a state-of-the-art batch selection method for image retrieval.

The **SIVAL dataset** contains 1500 images, each labeled with one of 25 object labels (e.g., gloves, apple). The cluttered images contain objects in a variety of poses and lighting conditions. We use the color and texture features provided by the dataset creators[3], which gives a 30-d descriptor for each of 30 regions per image. For this dataset, manual effort per image is the time required for manual segmentation; we use the cost data provided by [21], though our recent work shows that the costs can be predicted using image features alone [11].

The **Hollywood dataset** (HOHA) contains 444 video samples with human actions from 32 movies [26]. Each sample is labeled according to one or more of 8 action classes (e.g., AnswerPhone, GetOutCar, HandShake). We use the "clean" training set. For features, we use the authors' code to extract HoG-HoF descriptors around space-time Harris interest points, and convert each action clip into a bag-of-words, with 1000 words. We use the length of a video-clip to measure the annotation effort, since a human will watch the entire clip in order to identify which of the actions are performed in it. Table 1 shows the distribution of annotation costs on the two above datasets.

The **Corel dataset** contains 5,000 images from 50 different categories (e.g., antelope, butterfly, car, cat), as selected by the authors of [12]. Each category contains 100 images. We use the features provided on the authors' website[4], which consist of color moments, edge histograms, and a wavelet-based texture feature.

For SIVAL, we use an RBF kernel with $\gamma = 10^-5$, which we set based on the feature dimension. For HOHA, we use a χ^2 RBF kernel on HoG and HoF, with parameters as specified in [26]. We set the SVM penalty parameters as $C = 100$ and $C_u = 100$ for all approaches, a large value intended to emphasize correct classification of the selected examples. We train and test all approaches in the one-vs-all setting, and use the standard train-test splits. For SIVAL and HOHA, our active learner's initial training set consists of five positive and five negative images per class, selected at random; we use the remainder as the unlabeled pool. We average all results over five such random selections. Since the quality of the initial classifier varies across runs and categories, we omit error bars for clarity. Instead, please see the supplementary file for all individual runs. [5].

[3]http://www.cs.wustl.edu/accio/

[4]http://www.cais.ntu.edu.sg/∼chhoi/SVMBMAL/

[5]http://www.cs.utexas.edu/∼svnaras/papers/budget-supplementary.pdf

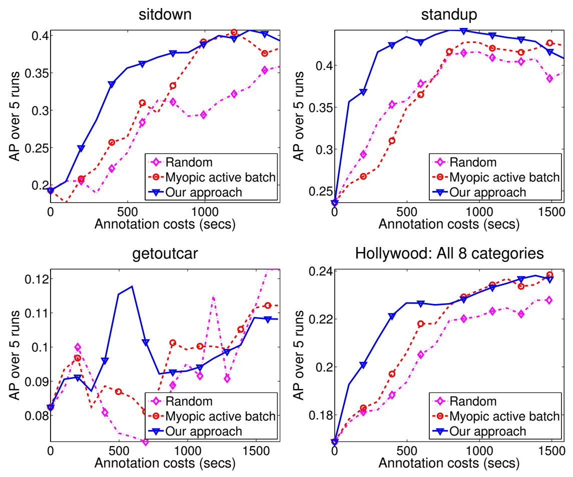

Figure 1. Results on the Hollywood dataset: example per-category learning curves (first two are best, third is worst) and the average results over all eight categories (bottom right plot) when actively learning categories of human activity from video clips.

Learning Activities on a Budget. Figure 1 shows representative (best and worst) learning curves for our method and the passive and myopic baselines plotted against the cost (annotation time) of the selected examples on the Hollywood dataset. The budget T is set such that all the unlabeled examples would be exhausted in about 20 batch iterations. About 10-15 examples on average get chosen per iteration. Note that average precision (AP) is plotted against the effort required to obtain annotations on the selected examples—not the number of queries—since the videos vary in length and require variable time to annotate.

All three methods steadily improve upon the initial classifier, but at different rates with respect to the cost. In general, a steeper learning curve indicates that a method is learning most effectively from the supplied labels. For most classes, our approach shows the most significant gains at a lower cost, meaning that it is best suited for maximally utilizing a budget. MAB is a bit better than random selection for most cases, but is weaker than our method due to its failure to account for the examples' annotation cost and potential redundancy. Our results on some actions (e.g., "get out of car") are more variable than others, which we attribute to the fact that the training and test clips are from distinct movies, and therefore vary a lot in terms of lighting, appearance, characters, etc. Overall, however, our approach consistently produces better accuracy for lower annotation cost, and outperforms the baselines on average over all eight actions (bottom right plot in Figure 1).

Learning Objects on a Budget. Figure 2 shows corresponding results on the SIVAL dataset. The budget T is set to 300 secs, again so that all unlabeled data would be exhausted in ∼20 iterations. Our approach is consistently better than both baselines, as seen in the bottom right plot.

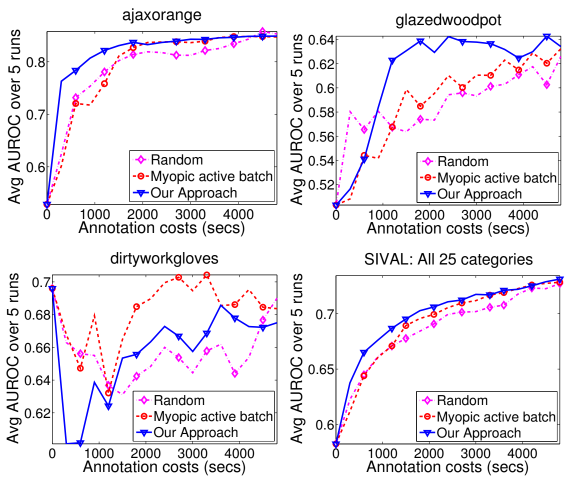

Figure 2. Results on the SIVAL dataset: example learning curves (first two are best, third is worst) and the average over all 25 categories (bottom right plot) when actively learning object categories.

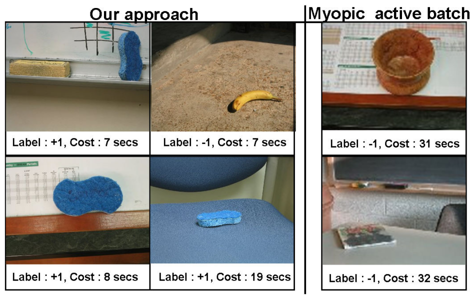
Our approach
Myopic active batch
Label : +1, Cost : 7 secs
Label : -1, Cost : 7 secs
Label : -1, Cost : 31 secs
Label : +1, Cost : 8 secs
Label : +1, Cost : 19 secs
Label : -1, Cost : 32 secs

Figure 3. Example batch selection made by our approach (left) and the myopic baseline (right) for the SIVAL "bluescrunge" object on the first iteration, with a budget of 60 secs.

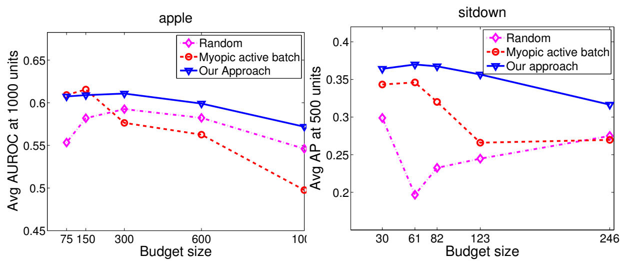

Figure 4. Active learning performance as a function of increasing budget size. The quality of our far-sighted selections remains more stable for larger budgets.

For some categories (such as "dirtyworkgloves"), none of the approaches improve with more labels, apparently due to those objects' non-descript texture/color. While the differences between the approaches may appear to be smaller that what we see for HOHA, they are consistent and significant considering that the results are averaged over five random initializations and 25 categories. Moreover, to achieve about 90% of the ultimate accuracy level possible on this dataset (0.7 AUROC), our method requires notably less cost: about 43% less annotation cost than the passive selector, and 20% less than the myopic selector.

Figure 3 shows an example batch selection made by our approach and the myopic baseline. The example illustrates the main advantage of our approach: we are able to select both less expensive and more informative examples, while sticking within the allowed budget as closely as possible.

Varying the Budget Size. Next we study the impact of increasing budget sizes. We expect the far-sightedness of our approach to offer particular advantages for larger budgets. For this experiment, we vary the size of the budget, and measure the accuracy of our method and the baselines at a fixed annotation cost for each budget (approx. $\frac{1}{4}$ of the total unlabeled pool's annotation cost). The range of budget sizes tested was set so as to exhaust all unlabeled data in about $10, \dots, 40$ iterations.

Figure 4 shows the results, for two example categories from SIVAL and HOHA (see supp. file for all classes). We include a minimal budget size to illustrate that for a budget allowing only ≈a single item to be selected, MAB and our approach would be almost equivalent (see leftmost points on both plots). As expected, for larger budgets, the myopic choices drop in accuracy, sometimes below the random baseline. Passive selection's accuracy is stable across budget sizes since it is simply random. Our approach shows the least degradation—a consequence of considering how the classifier changes if we were to obtain the most probable labels on the candidate examples for selection. This is a key result, given that real recognition systems drawing on a pool of annotators must be able to pick a large batch of jobs wisely in order to farm them out in parallel.

Comparison to State-of-the-Art Batch Selection. Next we provide comparisons with the state-of-the-art batch-mode active learning (BMAL) method of [12] on a CBIR task with Corel. The two BMAL variants use quadratic programming ($\text{SVM}_{BMAL}^{SS(QP)}$) and combinatorial optimization ($\text{SVM}_{BMAL}^{SS(CO)}$). While their approach is intended for fixed-size batches, and ours allows variable-sized batches, we can still test our method in this setting since it is a special case (i.e., budget=batch size). We replicate the experimental setup given by the authors, using 200 random queries, and applying the same kernel [27], SVM parameters, and scoring criteria (see [12] for details).

Table 2 shows the results. Our results are comparable, if not better, than the state-of-the-art, and the gains are a bit more apparent with larger batch sizes. We attribute our gains to our method's inclusion of the expected classifier change. (See [12] for more results from other active selection baselines (including [2, 14]), all of which generally underperform BMAL, and thus our method, for this data.)

Impact of Budgets Vs. Fixed-Size Batches. Finally, we examine the impact of being able to select variable-sized batches according to a fixed budget, as compared to fixing

Precision	Batch Size					
	5	10	15	20	25	30
Ours	0.620	0.734	0.809	0.853	0.888	0.905
$\text{SVM}_{BMAL}^{SS(QP)}$	0.640	0.718	0.798	0.835	0.860	0.886
$\text{SVM}_{BMAL}^{SS(CO)}$	0.622	0.717	0.776	0.835	0.868	0.889

Recall	Batch Size					
	5	10	15	20	25	30
Ours	0.321	0.371	0.417	0.452	0.477	0.503
$\text{SVM}_{BMAL}^{SS(QP)}$	0.332	0.373	0.423	0.452	0.468	0.490
$\text{SVM}_{BMAL}^{SS(CO)}$	0.321	0.377	0.412	0.447	0.471	0.493

Table 2. Corel results. **Top:** The average precision of the top 20 retrievals with different batch sizes. **Bottom:** The average recall of the top 100 retrievals with different batch sizes (evaluation done as prescribed in [12]).

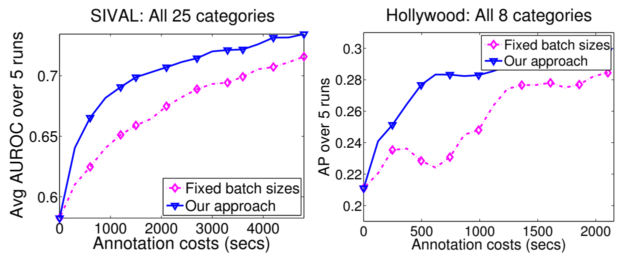

Figure 5. Comparison of active batch selections when using our budgeted approach, vs. restricting selections to a fixed batch size.

a batch size. We implement a QP-solver for the BMAL approach [12] and run experiments on SIVAL and HOHA. Since the BMAL baseline must choose k examples at each iteration (regardless of the annotation cost), we set k to the budget T divided by the dataset's mean cost.

Figure 5 shows the results. On both datasets, our budgeted selection performs better than a fixed-batch choice. This reinforces our claim that the higher the annotation cost variability among the unlabeled data, the more crucial it is to optimize selections for the given budget. Our method essentially picks a mixture of less/more expensive examples so as to best utilize the allowed annotation budget, whereas a method limited to choosing fixed-size batches is misled into choosing a seemingly informative batch that may be overly expensive in reality.

5. Conclusions

In conclusion, in this work we formalize the problem of far-sighted active learning on a budget, and propose a new method for optimally selecting a set of examples for a support vector machine classifier under these conditions. We provide an efficient iterative minimization technique that balances candidate examples' labeling costs and value when selected in batches. Experiments on three benchmark datasets show the practical advantages when compared to passive and myopic active alternatives, as well as an existing active batch selection baseline. Overall the results are quite encouraging and suggest that the proposed approach enables wise use of budgeted supervision.

Acknowledgements. This research is supported in part by NSF CAREER 0747356, NSF EIA-0303609, gifts from Google Research and Microsoft Research, DARPA VIRAT, and the Henry Luce Foundation.

References

[1] Y. Freund, H. Seung, E. Shamir, and Tishby. Selective Sampling the Query by Committee Algorithm. *Machine Learning*, 28, 1997.
[2] S. Tong and D. Koller. Support Vector Machine Active Learning with Applications to Text Classification. In *ICML*, 2000.
[3] M. Lindenbaum, S. Markovitch, and D. Rusakov. Selective Sampling for Nearest Neighbor Classifiers. *Machine Learning*, 54(2), 2004.
[4] A. Kapoor, E. Horvitz, and S. Basu. Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning. In *IJCAI*, 2007.
[5] C. Campbell, N. Cristianini, and A. Smola. Query Learning with Large Margin Classifiers. In *ICML*, 2000.
[6] E. Chang, S. Tong, K. Goh, and C. Chang. Support Vector Machine Concept-Dependent Active Learning for Image Retrieval. In *IEEE Transactions on Multimedia*, 2005.
[7] R. Yan, J. Yang, and A. Hauptmann. Automatically Labeling Video Data using Multi-Class Active Learning. In *ICCV*, 2003.
[8] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active Learning with Gaussian Processes for Object Categorization. In *ICCV*, 2007.
[9] G. Qi, X. Hua, Y. Rui, J. Tang, and H. Zhang. Two-Dimensional Active Learning for Image Classification. In *CVPR*, 2008.
[10] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards Scalable Dataset Construction: An Active Learning Approach. In *ECCV*, 2008.
[11] S. Vijayanarasimhan and K. Grauman. What's It Going to Cost You?: Predicting Effort vs. Informativeness for Multi-Label Image Annotations. In *CVPR*, 2009.
[12] S. Hoi, R. Jin, J. Zhu, and M. Lyu. Semi-supervised SVM Batch Mode Active Learning with Applications to Image Retrieval. *ACM Transactions on Information Systems*, 1(1), 2009.
[13] G. Schohn and D. Cohn. Less is More: Active Learning with Support Vector Machines. In *ICML*, 2000.
[14] K. Brinker. Incorporating Diversity in Active Learning with Support Vector Machines. In *ICML*, 2003.
[15] Y. Guo and D. Schuurmans. Discriminative Batch Mode Active Learning. In *NIPS*, 2007.
[16] K. Baldridge and M. Osborne. Active Learning and Logarithmic Opinion Pools for Hpsg Parse Selection. *Natural Language Engineering*, 14(2):191–222, 2008.
[17] R. Greiner, A. Grove, and D. Roth. Learning Cost-Sensitive Active Classifiers. *Artificial Intelligence*, 139(2):137–174, 2002.
[18] S. Vijayanarasimhan and K. Grauman. Multi-Level Active Prediction of Useful Image Annotations for Recognition. In *NIPS*, 2008.
[19] B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
[20] R. Haertel, E. Ringger, K. Seppi, J. Carroll, and P. McClanahan. Assessing the Costs of Sampling Methods in Active Learning for Annotation. In *Proceedings of ACL*, 2008.
[21] B. Settles, M. Craven, and L. Friedland. Active Learning with Real Annotation Costs. In *NIPS Wkshp on Cost-Sensitive Learning*, 2008.
[22] A. Sorokin and D. Forsyth. Utility Data Annotation with Amazon Mechanical Turk. In *CVPR Wkshp on Internet Vision*, 2008.
[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.
[24] A. Krause and C. Guestrin. Nonmyopic Active Learning of Gaussian Processes: an Exploration-Exploitation Approach. In *ICML*, 2007.
[25] T. Joachims. Transductive Inference for Text Classification using Support Vector Machines. In *ICML*, 1999.
[26] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. In *CVPR*, 2008.
[27] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the Point Cloud: from Transductive to Semi-supervised Learning. In *ICML*, 2005.