Hierarchical Subquery Evaluation for Active Learning on a Graph

Oisin Mac Aodha   Neill D.F. Campbell   Jan Kautz   Gabriel J. Brostow
University College London
http://visual.cs.ucl.ac.uk/pubs/graphActiveLearning

Abstract

To train good supervised and semi-supervised object classifiers, it is critical that we not waste the time of the human experts who are providing the training labels. Existing active learning strategies can have uneven performance, being efficient on some datasets but wasteful on others, or inconsistent just between runs on the same dataset. We propose perplexity based graph construction and a new hierarchical subquery evaluation algorithm to combat this variability, and to release the potential of Expected Error Reduction.

Under some specific circumstances, Expected Error Reduction has been one of the strongest-performing informativeness criteria for active learning. Until now, it has also been prohibitively costly to compute for sizeable datasets. We demonstrate our highly practical algorithm, comparing it to other active learning measures on classification datasets that vary in sparsity, dimensionality, and size. Our algorithm is consistent over multiple runs and achieves high accuracy, while querying the human expert for labels at a frequency that matches their desired time budget.

1. Introduction

Bespoke object recognizers are almost mature enough to be useful to people in practice. A major hurdle is how to procure enough training labels to tune a semi-supervised model for a specified classification task. While unskilled Mechanical Turkers are willing to label images of food at $1.40 per image [25], the costs are massive for recruiting and paying specialists like doctors or scientists. Whether they are experts or part of an online crowd, people need practical and reliable Active Learning (AL) to suggest which unlabeled image they, as the oracle, should label next. Choosing the query images in the right order gives better classification after fewer interrogations of the oracle.

During a training session, the classifier model starts with only unlabeled examples, picks one, queries the human for its label, and then quickly re-trains the classifier so the process can repeat with queries selected among the remaining unlabeled examples. We therefore work within the popular graph based semi-supervised learning (SSL) framework, where each image is represented as a vertex in a weighted graph, weights encode similarity between image feature vectors, and vertices that have already been queried have labels. Whether the human is done providing class labels or not, classification of all datapoints is performed directly in feature space by propagating available label information over the graph.

Designing a graph based AL framework requires three steps: 1) building a graph of the unlabeled datapoints in feature-space, 2) selection of an AL criterion for measuring the informativeness of possible queries, and 3) selecting an inference method for evaluating the criterion on the graph. There are many benefits to this framework, but forming the right combination of these three is an acknowledged challenge. The other steps are especially influenced by the AL criterion, chosen to decide which unlabeled image will be the next query. In particular, Expected Error Reduction (EER) is very attractive (see § 3.1), but naive incarnations of it are prohibitively costly. Each query put to the oracle is preceded by computing "subqueries" to each unlabeled example; a subquery simulates how the updated predictions *would* change if that individual datapoint received this or that label from the oracle.

We therefore propose a method for graph construction that is good in its own right, but crucially, organizes the data so that the EER criterion can be exploited effectively. Building on our graph construction, our main contribution is the proposed hierarchical subquery evaluation, which allows us to ask the oracle for a label that maximizes EER, without having to compute EER exhaustively for all unlabeled images, and without heuristics that hurt the overall learning curve. Our many experiments show that the significant benefits of computing EER by traversing our hierarchical representation of the data are 1) that we can cope with datasets having a broad variety of sparsity, dimensionality, and size, 2) that we balance exploration *vs*. exploitation to get good accuracy quickly and refine decision boundaries as needed within the time budget specified by the user, and 3) that empirically, we have highly consistent accuracy when labeling a given dataset. Our experiments benchmark our approach against alternative AL criteria and alternative graph constructions, and establish the repeatability of our approach across different datasets.

2. Related Work

Here we cover only the most relevant related works, and recommend [27] for a thorough overview of active learning. Active learning has been successfully applied to many different computer vision problems including tracking [32], image categorization [17], object detection [31], semantic segmentation [30], and image [1] and video segmentation [8], with both human and automatic oracles [18]. Compared to the body of work on active learning in general, there are relatively few active learning methods for image classification which facilitate *interactive* annotation. The challenge with creating interactive algorithms is that the time to retrain the model, once a labeled example is provided, can be long if not performed incrementally. This delay can also be further exacerbated by the type of active learning criterion used. Yao *et al.* [37] propose object detection based on efficient incremental training of Hough voting forests. Operating in real-time, their system is able to predict an annotation cost for an image and provides feedback to the user. However, they do not exploit the unlabeled data in the pool when updating their model. Batra *et al.* [1] present a system for interactive image co-segmentation which asks the user to annotate the region deemed to be most informative by the current model. Wang *et al.* [34] perform cell image annotation using a semi-supervised graph labeling approach and exploit fast updating of the graph for interactive annotations. Unlike our work, they do not explore the merits of different active learning criteria.

2.1. Semi-Supervised Active Learning

In pool based active learning we have access to the unlabeled data up front, before querying the oracle. In contrast to standard supervised learning, semi-supervised learning (SSL) exploits structure in the unlabeled data. In this paper we are concerned with graph based SSL, however our proposed subquery evaluation scheme can be applied to any pool based active learning task where the unlabeled data is available during training. In graph based SSL, datapoints are represented as nodes in a graph and edges between the nodes encode similarity in feature space. The premise is that datapoints near each other in feature space share the same label. Graph based transductive algorithms can be efficient to evaluate in closed form, typically only requiring simple matrix operations to propagate label information around the graph.

Graph Based SSL: Zhu *et al.* [40] propose an approach to SSL based on defining harmonic functions on Gaussian random fields. The advantage of their method is that, unlike graph cut based formulations [2], it produces a probability distribution over the possible class labels for each datapoint. Having real probabilities opens the door to a broader range of active learning strategies. The LGC method of Zhou *et al.* [38], adds additional regularization by balancing the information a node receives from the labeled set and its neighbors, but at the expense of allowing a labeled node to change class. For both methods it is also possible to include a label regularization term to address class imbalance in the data [34].

As the number of datapoints increases, it can quickly become infeasible to perform the large matrix inversions that are required by many graph based SSL algorithms. Iterative algorithms do not require a matrix inversion but can take many iterations to converge [39, 38]. Options to overcome this scalability issue include reducing the effective graph size using mixture models in feature space [41], nonparametric regression of the labels through a subset of anchor nodes [22], or assuming the data to be dimensionally separable in order to approximate the eigenvectors of the normalized graph Laplacian [10].

Graph Construction: It is well known that graph based methods are highly sensitive to the choice of edge weights [16]. A standard approach for graph construction is to first sparsify the fully-connected graph and then reweight the remaining edges. Sparsification is important, because in higher dimensions, the distances between far away points become less meaningful. K-nearest neighbor and distance thresholding are common choices for sparsification. However, they suffer from the problem that the resulting graph can be uneven as there is no guarantee on the number of edges at each node. Approaches exist to guarantee regular graphs (the same number of edges at each node) but can be computationally costly [16]. However, for a small decrease in graph quality, it is possible to build approximately regular graphs at reduced cost [36]. In the reweighting step, a similarity measure between datapoints must be defined. One standard choice of similarity is the RBF kernel, and several methods have been proposed to define a suitable bandwidth parameter. If there are labeled datapoints it can be learned [40], alternatively it can be defined per dimension, based on the average distance between all neighbors [4], local distance [12], or by direct optimization [33]. Wang *et al.* [35] jointly learn the graph structure and label prediction by minimizing a cost function over the graph and its labeling. In this paper we propose a method for graph reweighting inspired by ideas from dimensionality reduction [13].

Active Learning on Graphs: Many different active learning criteria exist in the literature. Methods range from random querying, uncertainty sampling, margin reduction, density sampling, expected model change, and expected error reduction [27]. An optimal strategy would trade off between exploration and exploitation; initially exploring the space when there are few labels and uncertainty is high and then, when more annotations have been acquired, exploit this information to perform boundary refinement between the classes. Algorithms that switch between density based and uncertainty sampling typically require hyperparameters that are dataset specific [3], however more complex approaches strive to do this automatically [20, 7]. Expected error reduction (EER) [26] performs this trade off naturally. Instead of measuring a surrogate, it seeks out datapoints that will make the overall class distributions on the unlabeled data more discriminative by attempting to reduce the model's future generalization error.

However, full EER requires $\mathrm{O}\big(N^2\big)$ operations to determine which example minimizes the expected error under the current model, where N is the size of the dataset. This complexity stems from needing to retrain the model for each of the N subqueries in the unlabeled pool to evaluate their expected error. Efficient update methods for some commonly known algorithms exist, *e.g.* in graph based SSL making full EER only feasible on small graphs. Zhu *et al.* [42] demonstrated the superior performance of EER over other active learning criteria when combining it with their Gaussian fields formulation [40], and this serves as one of our baselines.

Clustering Approaches: To cope with larger datasets, different approaches have been proposed to reduce the number of subqueries that must be evaluated. Strategies include only considering a subsample of the full data [26], or using the inherent structure of the data to limit influence and selection of subqueries [23]. Using the same manifold assumption as SSL, these methods cluster the data in its original feature space. Macskassy [23] explores graph based metrics, commonly used in community detection, to identify cluster centers (each assumed to contain the same class) that are then evaluated using EER. This is related to the hierarchical clustering method for category discovery of Vatturi [29]. However, by limiting subqueries to cluster centers, these clustering based approaches are unable to perform boundary refinement.

The hierarchical clustering, in [6], is used to define bounds on sampling statistics. Every one of their samples (a full query to the oracle) is randomly selected from a strict partition of a prespecified clustering (similar to a breadth first search) and only shares label information within its cluster. Our proposed method also uses a hierarchical representation, but differs as it uses the hierarchy for efficient sampling using EER, with the added advantages of graph based SSL, without sacrificing the ability to refine class boundaries.

3. Graph Based Semi-Supervised Framework

Here we review graph based SSL, and detail our innovations in § 4. In pool based learning, one has a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ where each $\mathbf{x}_i$ is a Q dimensional feature vector and $y_i \in 1, ..., C$ is its corresponding class label. We split $\mathcal{D}$ into two disjoint sets $\mathcal{D}_u$ and $\mathcal{D}_l$, corresponding to the sets of unlabeled and labeled examples. For active learning, the set of labeled examples is initially empty as only the oracle knows the values of each y_i. One can define a graph G with a set of vertices $\mathcal{V}$, corresponding to the pool of N examples in $\mathcal{D}$, and the set of edges is represented by a connectivity weight matrix $W \in \mathbb{R}^{N\times N}$. Each entry w_{ij} in W represents the similarity in some feature space between datapoints $\mathbf{x}_i$ and $\mathbf{x}_j$. Our goal is to estimate the distribution over the class labels for each of the nodes in the graph, $f_{ic} = P(y_i = c \,|\, \mathbf{x}_i)$. In matrix notation, these distributions, F, are represented as an $N \times C$ matrix, where each row is a different datapoint.

Zhu *et al.* [40] propose a method for semi-supervised learning based on Gaussian random fields and harmonic energy minimization (GRF). Their harmonic energy minimization can be computed in closed form using matrix operations on the graph Laplacian,

$$F_u = (D_{uu} - W_{uu})^{-1} W_{ul} Y_l, \tag{1}$$

where D is a diagonal matrix with entries $d_{ii} = \sum_j w_{ij}$. The matrices are split into labeled and unlabeled parts

$$W = \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix}, \text{ and } Y = \begin{bmatrix} Y_l \\ Y_u \end{bmatrix}. \tag{2}$$

Again using matrix notation, Y is the same size as F where all entries are set to 0 except where the oracle labels datapoint $\mathbf{x}_i$ with class c making $y_{ic} = 1$.

3.1. Expected Error Reduction

Let $P(y|\mathbf{x})$ be the unknown conditional distribution of output y over input $\mathbf{x}$, and $P(\mathbf{x})$ be the marginal input distribution. Taking the labeled data $\mathcal{D}_l$, we can produce a learner that estimates the class output distribution $\hat{P}_{\mathcal{D}_l}(y|\mathbf{x})$ for a given input $\mathbf{x}$. The expected error of such a learner is

$$\mathcal{E}_{\hat{P}_{\mathcal{D}_l}} = \int_{\mathbf{x}} L\left(P(y|\mathbf{x}), \hat{P}_{\mathcal{D}_l}(y|\mathbf{x})\right), \tag{3}$$

where we define $L(\cdot,\cdot)$ as a loss function that quantifies any error between the predicted output and the true value. In our learning problem, we consider multi-class classification tasks and use a 0/1 loss function

$$L\left(P(y|\mathbf{x}), \hat{P}_{\mathcal{D}_l}(y|\mathbf{x})\right) = \sum_{y=1}^{C} P(y|\mathbf{x}) \,\mathrm{I}\left[y \neq \hat{y}\right], \tag{4}$$

where $\hat{y} = \arg\max_y \hat{P}_{\mathcal{D}_l}(y|\mathbf{x})$ is the learner's MAP estimate of the class of $\mathbf{x}$, and $\mathrm{I}[\cdot]$ is a binary indicator function.

In the case of graph based SSL, we represent the marginal input distribution by the set of input samples $\{\mathbf{x}_i\}$

and evaluate the integral of (3) as a summation over this set to produce

$$\mathcal{E}_{\hat{P}_{\mathcal{D}_l}} = \sum_{i=1}^{N} \sum_{y_i=1}^{C} P(y_i|\mathbf{x}_i)\ \mathrm{I}\left[y_i \neq \hat{y_i}\right] \tag{5}$$

as the expected error. In practice, the true conditional distribution $P(y|x)$ is unknown, so we approximate it using the current estimate of the learner $\hat{P}_{\mathcal{D}_l}(y|x)$.

In the context of active learning, we would like to select the oracle's next query $(\hat{\mathbf{x}}_q, \hat{y}_q)$ from the unlabeled data $\mathcal{D}_u$, such that adding it to the labeled data $\mathcal{D}_l$ would result in a new learner with a lower expected error. This leads to a greedy selection strategy. First, we determine the expected error (or *risk*) for combinations of each unlabeled example $\mathbf{x}_q \in \mathcal{D}_u$ taking each possible label $y_q \in \{1..C\}$

$$\mathcal{E}^{+(\mathbf{x}_q,y_q)}_{\hat{P}_{\mathcal{D}_l}} = \sum_{i=1}^{N} \sum_{y_i=1}^{C} \hat{P}^{+(\mathbf{x}_q,y_q)}_{\mathcal{D}_l}(y_i|\mathbf{x}_i)\ \mathrm{I}\Big[y_i \neq \hat{y_i}^{+(\mathbf{x}_q,y_q)}\Big]\ , \tag{6}$$

where $\hat{P}^{+(\mathbf{x}_q,y_q)}_{\mathcal{D}_l}$ is the learner with $(\mathbf{x}_q, y_q)$ added to the labeled data. We then calculate the expectation of this risk across the possible label values for y_q. We use the learner's current posterior $\hat{P}_{\mathcal{D}_l}(y_q|x_q)$ to approximate the unknown true distribution across y_q to provide

$$\mathrm{E}\left[\mathcal{E}^{+(\mathbf{x}_q,y_q)}_{\hat{P}_{\mathcal{D}_l}}\right] = \sum_{y'=1}^{C} \hat{P}_{\mathcal{D}_l}(y_q{=}y' \,|\, \mathbf{x}_q)\ \mathcal{E}^{+(\mathbf{x}_q,y_q=y')}_{\hat{P}_{\mathcal{D}_l}} \tag{7}$$

as the expected risk. Finally, we select the query $\hat{\mathbf{x}}_q$ with the smallest expected risk. For the remainder of the paper, we refer to this expected risk as the *expected error* that the EER criterion seeks to minimize.

Zhu *et al.* [42] integrated active learning into their GRF framework by exhaustively calculating the expected error over all possible unlabeled nodes. Even with the proposed matrix update efficiencies of Zhu *et al.*, calculating the expected error for a datapoint is a linear operation and evaluating it over all unlabeled examples results in a time complexity of $\mathrm{O}\big(|\mathcal{D}_u|^2\big)$. This quadratic cost is prohibitively expensive as the dataset increases in size. We address this limitation using our proposed hierarchical subquery sampling approach presented in § 4.2.

4. Hierarchical Subquery Evaluation

Our method uses the EER active learning criterion while overcoming the expense of exhaustive sampling. It does this without sacrificing the desirable exploration/exploitation properties of EER, an issue with previous subsampling approaches. Before we discuss our hierarchical subquery search method, we first describe our graph construction technique that we have found to work well with the EER criterion and to be robust across a wide variety of datasets.

4.1. Perplexity Based Graph Construction

As noted previously, graph based SSL algorithms are very sensitive to the choice of similarity matrix W. If two datapoints $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same label, we want their corresponding affinity w_{ij} to be high, and if they are different we want it to be low. One popular choice of similarity kernel is the radial basis function (RBF),

$$w_{ij} = \exp(-\gamma_i \|\mathbf{x}_i - \mathbf{x}_j\|_2^2). \tag{8}$$

Here we use the L_2 distance, but other distances may be more appropriate depending on the data representation (*e.g.* histograms). We have now introduced a set of parameters γ_i that control the bandwidth of the kernel. A single choice of γ is unlikely to be optimal across the whole dataset. We want each γ_i to model the density of local space. Intuitively, we want a larger value of γ_i in dense regions of the feature space and a smaller value in more sparse regions. We now define our similarity based on a successful unsupervised technique from dimensionality reduction.

In Stochastic Neighbor Embedding (SNE) [13] the nonsymmetric similarity between points is represented as a conditional probability. w_{ji} can be interpreted as the probability that $\mathbf{x}_i$ would pick $\mathbf{x}_j$ as its neighbor assuming there is a Gaussian with variance σ_i^2 centered at $\mathbf{x}_i$, where $\gamma_i = 1/(2\sigma_i^2)$. We perform the same binary search as SNE to find the values of γ_i that best match a given level of *perplexity* (a measure of the effective number of local neighbors). The perplexity for a given choice of γ_i is defined as

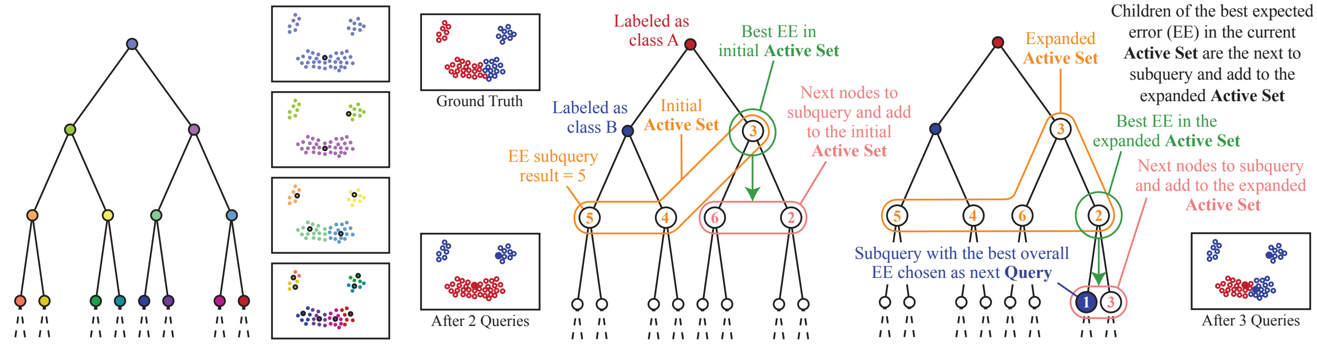
$$\mathrm{Perp}(\gamma_i) = 2^{-\sum_j w_{ji} \log_2 w_{ji}}\ . \tag{9}$$

We enforce a valid similarity matrix W by symmetrizing the conditional probabilities, so $w_{ji} = \frac{1}{2}(w_{ij} + w_{ji})$.

4.2. Hierarchical EER

The EER criterion dictates that we pick the datapoint giving the lowest expected error to be labeled next. We refer to calculating the expected error of a single unlabeled datapoint as a subquery; the complexity of a single subquery is linear in the number of unlabeled datapoints. Together, the subqueries are internal calculations used to determine the next query that is sent to the oracle for labeling. We want to find the next query within a specified query budget. This means we do not have sufficient time to perform subqueries on all possible unlabeled nodes since this results in a quadratic cost (§ 3.1). Instead, we must identify an adaptive number of the best subqueries to sample within an allotted time, ideally sub-linear in the number of unlabeled nodes.

The smooth nature of the harmonic solutions, with respect to proximity of nodes on the graph, creates a redundancy in densely sampling all nodes; neighboring nodes will likely produce a very similar reduction in error when labeled. A hierarchical clustering of the graph, for example


Ground Truth
After 2 Queries
Labeled as class A
Best EE in initial Active Set
Labeled as class B
Initial Active Set
EE subquery result = 5
Next nodes to subquery and add to the initial Active Set
3
5
4
6
2
Expanded Active Set
Children of the best expected error (EE) in the current Active Set are the next to subquery and add to the expanded Active Set
Best EE in the expanded Active Set
Next nodes to subquery and add to the expanded Active Set
Subquery with the best overall EE chosen as next Query
3
5
4
6
2
1
3
After 3 Queries

(a) The AuthorityShift clustering

(b) Our hierarchical subquery sampling algorithm for the next active learning query

Figure 1. Hierarchical clustering and subquery sampling strategy. (a) A hierarchical clustering is built using [5], shown here as a tree. At each level, every node in the tree is represented by a unique allocation (denoted by color) to a specific datapoint (the *authority point* in bold). (b) We use a hierarchical algorithm to determine the subqueries to perform; a subquery evaluates the expected error (EER criterion) shown as a number inside the node. (left) An *active set*, shown in orange, is constructed containing the children of labeled nodes; these are evaluated as the first subqueries, prioritizing from top to bottom. The active set is then expanded in a greedy fashion by including the children of the subquery with the lowest expected error, shown in pink. (right) We repeat this process until we have exhausted our subquery budget. The query for the oracle to label is chosen as the subquery with the lowest expected error (greatest EER).

Figure 1(a), exploits these local correlations between neighboring nodes. Previous approaches to reducing the number of subqueries have included random sub-sampling [26] and using community detection to propose candidates [23]. The latter method is equivalent to performing a breadth first (coarse to fine) search of a cluster hierarchy where graph communities are represented as high level clusters. Similar breadth first searches of hierarchies have been used in active learning, albeit without the EER criterion [6, 29].

The main advantage of the EER criterion is that it will trade-off the reduction in error achieved by either labeling an unknown region (exploration) or refining the decision boundaries under its current model (exploitation). Typically, the exploration mode will label nodes high up in the hierarchy whereas the detailed boundary refinement will occur in the leaves of the tree. While a breadth first approach can achieve good initial results, the active learner is stuck in an exploratory mode since it is effectively sampling on a graph density measure.

In our proposed approach, we allow the EER measure to perform the exploration/exploitation trade-off while still sub-sampling the unknown nodes to dramatically reduce the number of subqueries and therefore the cost. We achieve this by performing an adaptive search of the hierarchy.

4.3. Hierarchical Subquery Sampling

Authority-Shift Hierarchy Creation: We provide an illustrative example of the hierarchical clustering in Figure 1(a). We make use of the Authority-Shift algorithm of Cho and Lee [5]. It does not require a feature space but operates on the perplexity graph directly. This technique produces a hierarchical clustering on a graph by authority seeking: the process of allocating each node to a local 'authority' node (that represents the cluster). The calculation explores the steady state of a set of random walks on the graph at an appropriate scale. By increasing the scale parameter iteratively, a hierarchy of clusters can be built up to form a tree. This approach has two advantages. First, each cluster in the tree is represented by a specific datapoint that can be used to perform a subquery. Second, the clusters themselves encode walks on the graph under the same transition matrix used to evaluate the harmonic function, and therefore produce a summary of the results of calculating the expected error for all the datapoints in the cluster.

Subquery Sampling: An overview of our hierarchical sampling algorithm is provided in Figure 1(b). We differ from previous breadth first searching strategies by allowing an adaptive search on the tree to greedily seek for the minimum reduction in expected error. Referring to the diagram, consider a set of data with the cluster hierarchy of Figure 1(a), where two nodes have already been queried and labeled; see the left side of Figure 1(b). First, we build an *active set* of unlabeled nodes containing the children of labeled nodes, starting at the root. We proceed to perform a batch of subqueries of this active set (shown in orange) to obtain the expected error (the numbers inside the nodes). We then expand the active set by adding the children of the subquery in the current active set with the minimum expected error (shown in pink). As the children are added to the active set, they are evaluated as subqueries; see the right side of Figure 1(b). This process repeats until we have exhausted our budget of subqueries (a limit on the size of the active set). We now select the member of this active set with the minimal expected error as the next query to be labeled by the oracle. We prioritize the subquery evaluation by the level in the hierarchy (top-to-bottom) and then by ranking the nodes based on the total number of their descendants.

The boxes in Figure 1(b) provide a toy illustration of the

Dataset	N	D	C	Feat	rand	margin	entropy	RALF [7]	Zhu [42]	randS [26]	bFirst [23]	**HSE (ours)**
Glass [11]	214	10	6	-	0.732	0.605	0.599	0.763	**0.818**	0.810	0.782	0.804
Ecoli [11]	336	7	8	-	0.759	0.781	0.788	0.812	0.832	0.829	0.782	**0.833**
Segment [11]	635	18	7	-	0.811	0.717	0.680	0.832	**0.903**	0.896	0.840	0.896
FlickrMat [28]	1000	50	10	PCA BoW	0.172	0.131	0.125	0.242	**0.261**	0.244	0.249	0.259
Coil20 [24]	1440	20	20	PCA	0.558	0.392	0.456	0.713	0.729	0.757	0.756	**0.760**
LFW10 [15]	1456	50	10	PCA BoW	0.310	0.261	0.247	0.352	0.421	0.419	0.410	**0.422**
UIUCSport [21]	1579	50	8	PCA BoW	0.425	0.405	0.300	0.604	0.650	0.669	0.624	**0.671**
Gait [14]	2353	25	9	PCA	0.506	0.434	0.313	0.650	0.668	0.665	0.669	**0.696**
Oil [11]	3000	12	3	-	0.927	0.800	0.798	0.916	0.943	0.948	0.979	**0.986**
Caltech4 [9]	3188	20	4	PCA BoW	0.953	0.922	0.936	0.966	0.986	0.988	**0.993**	0.990
Eth80 [7]	3280	576	8	HoG	0.531	0.359	0.370	0.660	0.649	0.603	0.665	**0.675**
CpPascal08 [7]	4450	576	20	HoG	0.091	0.075	0.079	**0.277**	0.074	0.073	0.167	0.184
15Scenes [19]	4485	50	15	PCA BoW	0.255	0.236	0.144	0.548	0.535	0.505	0.469	**0.573**
Mean					0.541	0.471	0.449	0.641	0.651	0.647	0.645	**0.673**
Wins					0	0	0	1	3	0	1	**8**

Table 1. Datasets used for our evaluation where N, D, C, and Feat refer to the number of datapoints, dimensionality, number of classes, and representation. Results are presented as areas under the learning curve (1.0 is ideal). The learning curves for a subset of these datasets are depicted in Figure 2. Our method outperforms the other baselines, including full EER [42] despite requiring far fewer subquery evaluations.

advantage of this approach. To refine the boundary between the two classes, we need to ask the oracle to label nodes at the edges of clusters; these are usually found low down in the hierarchy. Because the EER improves as one moves toward a decision boundary, the active set can move down into the tree when the EER criterion favors exploitation over the improvement of exploration; exploration occurs by labeling clusters at the top of the tree. Under breadth first search, a large number of queries would have to be performed before reaching nodes at the exploitation depth. As the learning curve evolves, the boundary refinement nodes will become increasingly localized, making it more unlikely that they will be found by random subqueries alone. We always take the root node of the tree as our first query (an open question for many algorithms) which we observe empirically to confer good performance and makes our algorithm deterministic. The tree construction means that the entire hierarchy has the potential to be navigated in $\mathrm{O}\big(Nlog(N)\big)$.

5. Experiments

Table 1 describes the 13 vision and standard machine learning datasets used for our experiments. These were chosen because they vary in size, density in their respective feature spaces, and have different numbers of classes. For all experiments, we start out with 3 random queries, construct graphs with 10 nearest neighbors based on the L_2 distance, use a perplexity value of 30, and query the oracle 50 times. For our method (HSE), we set the number of subqueries to be $25\log(N)$, where N is the number of datapoints for a given dataset, and the initial queries are set as the first 3 nodes in the hierarchy. Data and code are available on our project webpage.

Graph Construction: Graph based SSL algorithms can produce inferior performance with poor graphs. Using the method of Zhu *et al.* [42] to evaluate graphs, Table 2 compares our perplexity based graph construction method to four other baseline algorithms, testing this contribution in isolation. For *mean*, the bandwidth of the RBF kernel is set using the average distance between neighbors. For *binary*, we set a constant value for any two nodes that are connected and zero elsewhere. For *knn*, the bandwidth is set per datapoint proportional to its K-nearest neighbors. Finally, *lle* is the local linear embedding approach of [33]. Our perplexity based graph performs best overall.

Active Learning Criteria: We compare our algorithm to seven different baselines, including GRF [40] with random, entropy, and margin based criteria [27], full EER [42], and the recent time varying combination approach RALF [7]. We also compare to two different subquery evaluation strategies, random [26] and breadth first [23]. Both competing subquery strategies are evaluated using the same number of subqueries as our method. All methods use our perplexity based graph with the exception of RALF which uses a binary based graph representation. Empirically, we found RALF to perform worse using other graphs. Table 1 summarizes our overall results as area under the learning curve on the unlabelled set.

Interestingly, our method outperforms full EER which requires $\mathrm{O}\big(N^2\big)$ computations. We note that the full EER is still a greedy algorithm at each iteration and therefore, not necessarily globally optimal. Our approach will encourage exploration at the start, when only a few queries have been performed and the active set is at the top of the hierarchy, which is observed to offer improved performance.

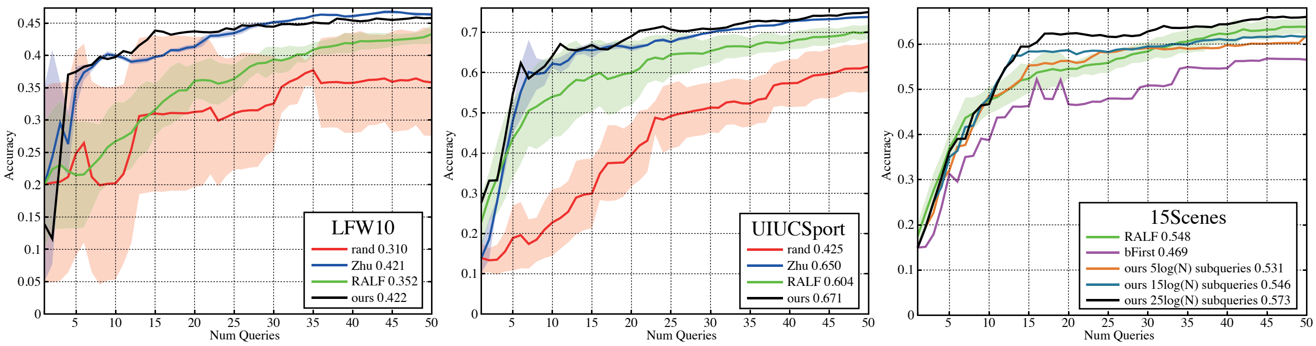


Figure 2. Learning curves illustrating the performance of our approach versus three other baselines from Table 1. The shaded regions around each learning curve represents one standard deviation. Our method gives superior results compared to that of Zhu *et al*. [42] and as it is deterministic, results do not vary over different runs. In the last plot we illustrate the effect of increasing the number of subqueries for our method. As the number increases, so does the area under the curve.

Dataset	mean	binary	knn [12]	lle [33]	**per (ours)**
Glass	0.775	0.743	0.758	0.787	**0.818**
Ecoli	0.795	0.768	0.777	0.791	**0.832**
Segment	0.837	0.860	0.853	0.892	**0.903**
FlickrMat	0.196	0.159	0.198	0.222	**0.261**
Coil20	0.641	0.597	0.616	**0.729**	**0.729**
LFW10	0.362	0.356	0.365	0.381	**0.421**
UIUCSport	0.528	0.452	0.527	0.529	**0.650**
Gait	**0.686**	0.646	0.672	0.579	0.668
Oil	0.941	0.937	0.924	**0.962**	0.943
Caltech4	0.981	0.973	0.977	0.971	**0.986**
Eth80	0.572	0.596	0.562	0.604	**0.649**
CpPascal08	0.146	0.102	**0.159**	0.141	0.074
15Scenes	0.344	0.304	0.353	0.378	**0.535**
Mean	0.600	0.576	0.595	0.613	**0.651**
Wins	1	0	1	2	**10**

Table 2. Comparison of different graph construction methods. The results represent area under learning curves for the GRF method of Zhu *et al*. [42]. Our perplexity based method outperforms the other baselines.

One noticeable exception is the Cropped Pascal dataset from [7]. Due to the high variability in each class, it is likely that this dataset does not conform to the clustering assumption of semi-supervised learning. Using an iterative label propagation algorithm with few propagation steps prevents RALF [7] from overfitting the dataset at the expense of worse marginals. Figure 2 illustrates learning curves for a subset of the datasets.

Table 3 depicts the average time required to present the next query to the user for the different active learning methods. RALF [7] scales linearly while full EER [42] soon becomes impractical as the the number of examples increases. On average, our method computes queries in under a second and performs better than both methods in terms of accuracy.

Dataset	RALF [7]	Zhu [42]	**HSE (ours)**
Glass	0.003	0.008	0.291
Ecoli	0.004	0.016	0.302
Segment	0.005	0.056	0.276
FlickrMat	0.007	0.231	0.136
Coil20	0.011	0.950	0.369
LFW10	0.009	0.535	0.172
UIUCSport	0.009	0.507	0.172
Gait	0.012	1.610	0.257
Oil	0.010	1.008	0.339
Caltech4	0.011	1.435	0.351
Eth80	0.014	2.793	0.378
CpPascal08	0.041	12.189	0.753
15Scenes	0.033	9.405	0.710

Table 3. Average time (in seconds) per query for active learning methods with different area-under-learning curve and across datasets of varying complexity. Both RALF and HSE pick the next query in under a second. In our method, we allow $25\log(N)$ subqueries per query rather than the full N^2 required for the Zhu method.

6. Discussion

Accurate AL is the key to saving human effort, but speed is also a factor when a human oracle's patience is finite. Generalizing slightly, our Active Learning approach performs as accurately or better than Zhu *et al*. [42], but does so with an effective computational complexity on par with Ebert *et al*. [7]. Their computational complexities are $\mathrm{O}\big(N^2\big)$ and $\mathrm{O}\big(N\big)$ respectively, while ours is $\mathrm{O}\big(N\log(N)\big)$ with a low $\log(N)$. In practice, with our Matlab implementation and default settings (used throughout), the combined subqueries needed to pick the oracle's next query finished in under a second, even for the largest datasets tested. For bigger datasets, users may opt to use our algorithm with fewer subqueries to keep the labeling interactive. Both those main competitors are very good, excelling on specific datasets. Therefore it is important that validation of our AL approach has considered accuracy, efficiency, and generalizability to a variety of situations. The online supplementary material further illustrates that across these datasets, our hierarchical subquery evaluation leads to accurate results in the form of steep learning curves with large areas under the curve, and that these results are consistent across multiple runs, as plotted with ±1 standard deviation from each curve's mean.

To tease apart the impact of our hierarchical subquery evaluation *vs*. our perplexity-based graph construction, we gave our graphs to the compatible AL baseline algorithms. Zhu *et al*. is among them, and without our graphs, performs worse than RALF. Within the flexible graph based SSL framework, other choices can also have an impact, so as part of the supplemental files, we also show that LGC, used by RALF, is not as effective for our label propagation as Zhu *et al*.'s GRF.

There are several exciting avenues for future work. Our approach is transductive, so it would be attractive to either embed new datapoints into our existing graph online, or to transfer learned parameters to an inductive model. It would also be interesting to budget subqueries to account for some labels taking more of the oracle's time or effort than others. Finally, our similarity graph is computed once offline and never updated. In future, we may wish to use the label information from the user to learn a feature representation online.

Acknowledgements: Funding for this research was provided by EPSRC grants EP/K015664/1, EP/J021458/1 and EP/I031170/1.

References

[1] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. iCoseg: Interactive co-segmentation with intelligent scribble guidance. *CVPR*, 2010.

[2] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. *ICML*, 2001.

[3] N. Cebron and M. R. Berthold. Active learning for object classification: from exploration to exploitation. *Data Mining and Knowledge Discovery*, 2009.

[4] O. Chapelle, B. Schölkopf, A. Zien, et al. *Semi-supervised learning*. MIT press Cambridge, 2006.

[5] M. Cho and K. MuLee. Authority-shift clustering: Hierarchical clustering by authority seeking on graphs. *CVPR*, 2010.

[6] S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. *ICML*, 2008.

[7] S. Ebert, M. Fritz, and B. Schiele. RALF: A reinforced active learning formulation for object class recognition. *CVPR*, 2012.

[8] A. Fathi, M. F. Balcan, X. Ren, and J. M. Rehg. Combining self training and active learning for video segmentation. *BMVC*, 2011.

[9] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *PAMI*, 2006.

[10] R. Fergus, Y. Weiss, and A. Torralba. Semi-supervised learning in gigantic image collections. *NIPS*, 2009.

[11] A. Frank and A. Asuncion. UCI machine learning repository. 2010.

[12] M. Hein and M. Maier. Manifold denoising. *NIPS*, 2006.

[13] G. Hinton and S. Roweis. Stochastic neighbor embedding. *NIPS*, 2002.

[14] T. Hospedales, S. Gong, and T. Xiang. Finding rare classes: Active learning with generative and discriminative models. *TKDE*, 2013.

[15] G. B. Huang, M. Mattar, H. Lee, and E. G. Learned-Miller. Learning to align from scratch. *NIPS*, 2012.

[16] T. Jebara, J. Wang, and S.-F. Chang. Graph construction and b-matching for semi-supervised learning. *ICML*, 2009.

[17] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. *CVPR*, 2009.

[18] V. Karasev, A. Ravichandran, and S. Soatto. Active Frame, Location, and Detector Selection for Automated and Manual Video Annotation. *CVPR*, 2014.

[19] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006.

[20] X. Li and Y. Guo. Adaptive active learning for image classification. *CVPR*, 2013.

[21] Li, Li-Jia and Fei-Fei, Li. What, where and who? classifying events by scene and object recognition. *CVPR*, 2007.

[22] W. Liu, J. He, and S.-F. Chang. Large graph construction for scalable semi-supervised learning. *ICML*, 2010.

[23] S. A. Macskassy. Using graph-based metrics with empirical risk minimization to speed up active learning on networked data. *KDD*, 2009.

[24] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-20). 1996.

[25] J. Noronha, E. Hysen, H. Zhang, and K. Z. Gajos. Platemate: Crowdsourcing nutrition analysis from food photographs. *UIST*, 2011.

[26] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. *ICML*, 2001.

[27] B. Settles. *Active Learning*. Morgan & Claypool, 2012.

[28] L. Sharan, R. Rosenholtz, and E. Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 2009.

[29] P. Vatturi and W.-K. Wong. Category detection using hierarchical mean shift. *KDD*, 2009.

[30] A. Vezhnevets, J. M. Buhmann, and V. Ferrari. Active learning for semantic segmentation with expected change. *CVPR*, 2012.

[31] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *CVPR*, 2011.

[32] C. Vondrick and D. Ramanan. Video annotation and tracking with active learning. *NIPS*, 2011.

[33] F. Wang and C. Zhang. Label propagation through linear neighborhoods. *TKDE*, 2008.

[34] J. Wang, S.-F. Chang, X. Zhou, and S. Wong. Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels. *CVPR*, 2008.

[35] J. Wang, T. Jebara, and S.-F. Chang. Graph transduction via alternating minimization. *ICML*, 2008.

[36] J. Wang and Y. Xia. Fast graph construction using auction algorithm. *UAI*, 2012.

[37] A. Yao, J. Gall, C. Leistner, and L. Van Gool. Interactive object detection. *CVPR*, 2012.

[38] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. *NIPS*, 2004.

[39] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report MSU-CSE-00-2, School of Computer Science, CMU, 2002.

[40] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. *ICML*, 2003.

[41] X. Zhu and J. Lafferty. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. *ICML*, 2005.

[42] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. *ICML workshops*, 2003.