Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

# Active Learning from Relative Queries

**Buyue Qian*, Xiang Wang*, Fei Wang†, Hongfei Li†, Jieping Ye‡, and Ian Davidson***
*University of California, Davis, CA 95616 {byqian, xiang, indavidson}@ucdavis.edu
†IBM T. J. Watson Research, Yorktown Heights, NY 10598 {fwang, liho}@us.ibm.com
‡Arizona State University, Tempe, AZ 85287 jieping.ye@asu.edu

## Abstract

Active learning has been extensively studied and shown to be useful in solving real problems. The typical setting of traditional active learning methods is querying *labels* from an *oracle*. This is only possible if an expert exists, which may not be the case in many real world applications. In this paper, we focus on designing *easier* questions that can be answered by a non-expert. These questions poll relative information as opposed to absolute information and can be even generated from side-information. We propose an active learning approach that queries the *ordering* of the importance of an instance's neighbors rather than its label. We explore our approach on real datasets and make several interesting discoveries including that querying neighborhood information can be an effective question to ask and sometimes can even yield better performance than querying labels.

## 1 Introduction and Motivation

Active learning extends machine learning by allowing learning algorithms to typically query the labels from an oracle for currently unlabeled instances. Though enormous progress has been made in the active learning field in recent years, traditional active learning does not cover the scenarios where only "non-expert" advice is available. Consider the setting in Figure 1, though an expert could answer the absolute question of which class the query galaxy belongs to, an easier relative question that everyone can provide their opinion on is to rank order galaxies 2 and 3 w.r.t. the visual similarity to galaxy 1. The purpose of this work is to explore active learning in such context, *when labels are difficult to query and obtain*. We propose that rather than ask for exact labels we query to better understand the *neighborhood structure* of the instances.

**Problem Setting.** In this work we explore active learning with non-expert guidance in the context of the popular label propagation type of algorithms. In this class of algorithms each instance is a node in a graph and has a weighted neighbor set which are collectively used to propagate labels to the unlabeled points. Popular semi-supervised label propagation approaches include `GFHF` [Zhu *et al.*, 2003a] and `LGC` [Zhou *et al.*, 2003]. Therefore, the neighborhood structure (which instances are neighbors of each other and their similarity) is important for the performance of these algorithms since this determines where the labels are propagated. In our formulation the neighborhood structure is learnt by minimizing the reconstruction error of writing a point as a linear combination of its nearest neighbors. The given labels are then propagated to the unlabeled points which are then further propagated and so on for an infinite number of steps.

Galaxy 1

(a) Absolute: What class is Galaxy 1?

Galaxy 1 Galaxy 2 Galaxy 3

(b) Relative: Does Galaxy 2 or 3 look more similar to 1?

Figure 1: Absolute and relative questions on Galaxy Zoo data

**Proposal.** Our query strategy is to, rather than querying an instance's label, ask a non-expert to place an ordering (or a partial ordering) on the similarity of the neighbor set to the instance they are neighbors of. Since our active learning scheme is performed on a neighbor set, we focus on selecting the most important neighbor sets which we cast as a counting set cover problem. Using counting set cover we aim to locate the neighborhood which is most influential in the graph. In practice, our algorithm will iteratively select the most "informative" neighbor set for querying, and the advice from non-experts will be enforced as constraints in the subsequent re-learning of the neighborhood weights which are then used to help better propagate labels with the process being repeated.

It is important to note that in our method new labels are **not** added, rather the neighborhood weights are better estimated.

The primary benefit of querying neighborhood weights or structure is that the questions are easier to answer. This is useful as labels are expensive to query in many specialized domains where labeling an instance requires proficient domain knowledge, such as annotating a galaxy image or predicting a person's mental health condition from a brain MRI scan, two applications we shall focus on. The focus of this paper is the setting where comparisons between instances is possible. We explore one such setting - images - since it covers a huge range of possible applications, but other settings are also possible if the neighborhood structure can be presented in a meaningful way to the non-experts. The proposed algorithm is computationally efficient and can be easily parallelized as discussed later. The promising empirical results demonstrate the effectiveness of the proposed approach, and validate our idea of querying neighborhood structure. Moreover, our results indicate that in some cases querying neighborhood orderings can yield greater learning accuracy than using the same number of queries of labels.

**Contribution.** Our work makes several contributions. (1) We investigate a new form of knowledge injection, and to our knowledge is the first paper that actively queries the neighborhood structure. (2) Our method can query both labeled and unlabeled instances. (3) The approach is scalable to large problems since it divides the problem into a series of small problems each of which could be easily solved using quadratic programming. (4) We empirically show that crowdsourcing can be a legitimate non-expert source in our method.

## 2 Related Work

According to a recent survey on active learning [Settles, 2009], existing active learning algorithms can be summarized into six categories based on the objective of the query selection. However, all of them are label focused and hence are not directly comparable to our work.

*Uncertainty sampling* queries the instance about whose label the learning model is least confident [Lewis and Gale, 1994; Culotta and McCallum, 2005] while *Query by committee* queries the instance about whose label the committee members (classifiers) most disagree [Muslea *et al.*, 2000; Melville and Mooney, 2004]. The *Expected model change* query focus is on the instance that would impart the greatest change to the learning model [Settles *et al.*, 2008]. The *Expected risk Reduction* approach queries the instance which would minimize the expected future classification risk [Roy and Mccallum, 2001; Guo and Greiner, 2007; Kapoor *et al.*, 2007] whilst the *Variance Reduction* query strategy chooses the instance which would minimize the output variance such that the future generalization error can be minimized [Zhang and Oles, 2000]. Finally *Density-weighted method* queries the instance which is not only uncertain but also representative of the underlying distribution of data [Settles and Craven, 2008]. However, these approaches only focus on one aspect of active learning – the query strategy, and the other aspect of active learning – the design of questions – is not addressed.

A new direction in active learning is batch mode active learning [Hoi *et al.*, 2006; Chattopadhyay *et al.*, 2012] which asks the oracle a set of labels instead of a single label at a time. Although this is a more efficient querying method, it still requires the human experts to provide labels of a batch of instances and does not make the question itself mode efficient or easier. A novel direction proposed by [Rashidi and Cook, 2011] is a method that aggregates multiple instances into a generic active learning query based on rule induction, and has been empirically demonstrated to perform more effectively and efficiently than querying labels. However, since it is a rule-based learning algorithm, its usefulness is limited to the cases that the data is represented in a low dimensional space and every feature has to be interpretable. Additionally, though a generic question, it is still an absolute question as it requires human experts to have even stronger background knowledge than just querying labels. In contrast, we focused on designing a relative active learning query which could be answered by people without domain knowledge. [Tamuz *et al.*, 2011] explores using triplet-based relative-similarity queries to improve the learning of kernel, but the PSD requirement of kernel limits the neighborhood relations to be symmetric. [Wauthier *et al.*, 2012] presents an active spectral clustering algorithm that queries pairwise similarity, our work differs from this not only in learning setting (semi-supervised versus unsupervised) but also we do not require the users to provide a real-valued pairwise similarity, as they do, rather just some orderings between the instances in the neighbor set.

## 3 Active Learning from Neighbor Ordering

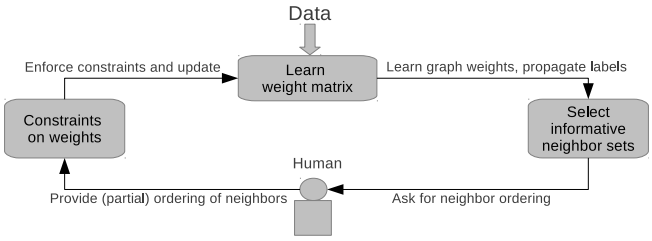



Figure 2: The cycle of the propose ALNO approach.

Figure 2 shows the major components and work flow of our proposed approach which we call active learning from neighbor ordering (ALNO). In the first step, the learning algorithm takes the data, learns the graph weights, and propagates the known labels to fill in missing labels. Then the most "informative" neighbor set is identified by solving a weighted counting set cover problem. In the next step of the learning cycle, the non-expert source (who could be a human or some other information source) is asked to place an ordering (or a partial ordering) on the identified neighbor set based on its similarity to the instances they are neighbors of. This neighborhood ordering information will later be encoded as constraints on the graph weights for the next iteration of learning.

**Notations.** Formally, the problem that we are trying to address is described as follows. Given a set of $n$ instances $X = \{x_1, x_2, \cdots, x_n\}$, we define an one-against-all classification problem on a set of $m$ possible labels. Let $Y \in \mathbb{R}^{n \times m}$

denote the prior (incomplete) label matrix, where $y_{ij} = 1$ if instance $x_i$ is labeled as class $j$, and $y_{ij} = 0$ otherwise. Let $\mathcal{N}_{x_i}$ denote neighbor set which consists of the nearest neighbors of $x_i$. Note that the size of the neighbor set may differ amongst instances. Then the graph weight matrix $W$ is learnt with an entry $w_{ij}$ indicating the "similarity" of instance $x_j$ (as a neighbor of $x_i$) to instance $x_i$. Note that we do not require the notion of similarity defined in $W$ to be symmetric, i.e., $w_{ij} \neq w_{ji}$ is allowed. This paper will often refer to row and column vectors of matrices, for instance, $i$-th row and $j$-th column vectors of $W$ are denoted as $W_{i\bullet}$ and $W_{\bullet j}$, respectively. In practice, our proposed approach iterates between the learning step – predicting the missing labels in $Y$ using a classifying function $F$, and the querying step – selecting informative neighbor sets and asking non-experts for the orderings. The major difference of our approach from traditional active learning methods is that instead of querying labels, we ask for an easier relative question – neighborhood ordering. We adopt the linear neighborhood propagation (LNP) algorithm proposed in [Wang and Zhang, 2006], which learns the graph weights $W$ by solving the reconstruction error as a quadratic program (QP). We shall in the next subsection briefly review the LNP method.

### 3.1 Background - LNP

**Learning of weights.** As introduced by [Roweis and Saul, 2000], the reconstruction error is defined as:

$$\mathcal{Q}(W) = \sum_{i=1}^{n} \|x_i - \sum_{j \in \mathcal{N}_i} w_{ij} x_j\|^2 \tag{1}$$

The reconstruction weight $W$ is typically solved as a constrained least square problem or a linear system of equations, however, [Wang and Zhang, 2006] have shown that it also can be solved as a QP. The advantage of using a QP formulation is that additional constraints (such as neighborhood ordering) can be added in, and thereby enables more flexibility to the formulation. Let $\mathcal{C}^i$ denote the local covariance matrix of instance $x_i$ (the term "local" refers to the fact that the instance is used as the mean in the calculation of covariance), formally the definition of $\mathcal{C}^i$ can be expressed as $\mathcal{C}^i = (\mathbf{1}x_i - \mathcal{N}_{x_i})(\mathbf{1}x_i - \mathcal{N}_{x_i})^T$, where $\mathbf{1}$ denotes a column vector consisting of ones. Using the local covariance matrix, the reconstruction error problem can be formulated as a series of small QP problems (one for each instance) since each row in $W$ is independent of every other:

$$\begin{aligned} \min_{W_{i\bullet}} \quad & W_{i\bullet}\mathcal{C}^i W_{i\bullet}^T \\ s.t. \quad & W_{i\bullet}\mathbf{1} = 1; \\ & w_{ij} \geqslant 0. \end{aligned} \tag{2}$$

**Label inference.** Once we have learnt the matrix $W$ we can use it to propagate labels from the labeled points to the unlabeled points. Since each row of the weight matrix $W$ sums to one, $W$ can be readily used as a transition matrix and perform a random walk on the graph to infer the missing labels. In each propagation iteration, the state (i.e. predicted label) of each data instance "absorbs" a portion ($\mu$) of the label information from its neighborhood, and retains a portion $(1-\mu)$ of its initial label information. Therefore, the state at time $t+1$ can be calculated using the previous state at time $t$.

$$F^{t+1} = \mu W F^t + (1-\mu)Y \tag{3}$$

Such a process will eventually converge to the following steady-state probability.

$$F^{\infty} = (1-\mu)(I - \mu W)^{-1} Y \tag{4}$$

### 3.2 Encoding Neighborhood Structure

As we exploit the acquired neighborhood orderings information, the first question we need to address would be how to enforce such ordering information into the learning process. The QP formulation of the reconstruction error minimization allows the encoding of the neighbor orderings as a set of constraints on the weights $W$. Here we take a simplified example to show how to enforce an ordering of two neighbors of an instance: Assume an instance $x_i$ has two neighbors $x_a$ and $x_b$, if instance $x_a$ is more similar to $x_i$ than $x_b$ to $x_i$, we can conclude that the weight of $x_a$ used to reconstruct $x_i$ is greater than that of $x_b$, i.e. $w_{ia} \geqslant w_{ib}$. This ordering can be encoded using the constraint as shown below.

$$W_{i\bullet}(\mathbf{J}^a - \mathbf{J}^b) \geqslant 0 \tag{5}$$

where $\mathbf{J}^i$ is a single-entry column vector whose $i$-th entry is one and all other entries are zeros. We can exploit transitivity to encode a complete ordering on a set of neighbors using multiple constraints. For example, if we require $w_{ia} \geqslant w_{ib} \geqslant w_{ic}$ this can be enforced using a pair of constraints, i.e., $W_{i\bullet}(\mathbf{J}^a - \mathbf{J}^b) \geqslant 0$ and $W_{i\bullet}(\mathbf{J}^b - \mathbf{J}^c) \geqslant 0$.

### 3.3 Query Selection as Counting Set Cover

We begin this section with the overview of the intuition behind the proposed approach and then providing full details for reproducibility of results. Our work deviates from existing active learning by querying not an instance's label, rather querying the neighborhood structure. Hence, our query strategy aims to choose neighbor sets which if queried will have the most impact in terms of better propagating the given labels on the graph. Since each neighbor set naturally forms a subset of the $n$ instances, we propose using counting set cover to estimate the importance of neighborhood sets. Benefits of using such a method include (i) neighbor sets that are essential to construct the graph are naturally captured via solving a set cover problem, and (ii) different weighting schemes used in the counting emphasize different notions of importance thus enriches the flexibility of active learning.

Recall a set cover problem consists of two parts: (1) A universe which in our case is the instance set $X$; (2) A set of subsets of $X$ which in our case is the $n$ neighbor sets, i.e., $\mathcal{N} = \{\mathcal{N}_{x_1}, \mathcal{N}_{x_2}, \cdots, \mathcal{N}_{x_n}\}$. We say a subset $\mathcal{S}$ ($\mathcal{S} \subset \mathcal{N}$) is a cover of the universe $X$ if every element in $X$ appears at least once in $\mathcal{S}$, i.e., $\cup \mathcal{S}_i = X$. A cover $\mathcal{S}$ that has minimum cardinality is called a minimum set cover. The set cover problem, which aims to identify such a minimum cover, can be formulated as an integer program.

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{n} \mathcal{Z}_i \\ s.t. \quad & \textstyle\sum_{j=1}^{n} \alpha_{ij} \mathcal{Z}_j \geq 1, \qquad \forall\, x_i \in X \\ & \mathcal{Z}_i \in \{0,1\} \end{aligned} \tag{6}$$

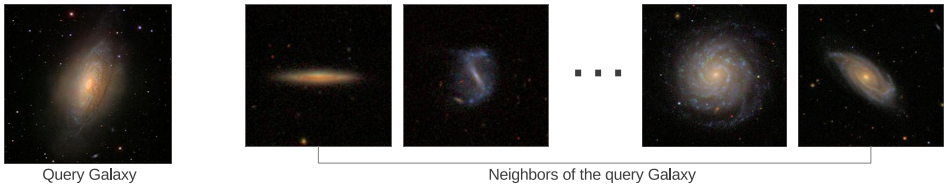



Figure 3: An example query in Galaxy Zoo data: (partially) order the neighbors based on their visual similarities to the query Galaxy.

where $\mathcal{Z}_i$ is set to be 1 iff the neighbor set $\mathcal{N}_{x_i}$ is part of a minimum set cover $\mathcal{S}$, and $\alpha_{ij} = 1$ if $x_i \in \mathcal{N}_{x_j}$ and $\alpha_{ij} = 0$ otherwise. Set cover is a well studied NP-hard problem. In our experiments we use an approximation algorithm, which involves a linear program relaxation of the original integer program (replacing the last constraint in Eq.(6) by $\mathcal{Z}_i \geq 0$) and then performing a randomized rounding. This provides a $2 \log n$ approximation with a probability of $(1 - \frac{1}{n})$ [Vazirani, 2001]. Such a method can produce multiple "close to minimum" set covers $\mathcal{S}^*$, hence we can count the number of solutions that each neighborhood $\mathcal{N}_{x_i}$ participates in to estimate its querying importance $\gamma(\mathcal{N}_{x_i})$. Formally, the importance counting is performed using:

$$\gamma(\mathcal{N}_{x_i}) = \sum_{\mathcal{S} \in \mathcal{S}^*} \delta(\mathcal{N}_{x_i}, \mathcal{S}) e(\mathcal{N}_{x_i}) \tag{7}$$

where $\mathcal{S}^*$ denotes the collection of multiple minimal set covers; $\delta(\mathcal{N}_{x_i}, \mathcal{S})$ is an indicator that takes value 1 if $\mathcal{N}_{x_i}$ is part of a minimum set cover $\mathcal{S}$, and takes value 0 otherwise. Here $e(\mathcal{N}_{x_i})$ is the weight of $\mathcal{N}_{x_i}$ used in the counting, which can be interpreted as our preference for querying the neighborhood ordering of $\mathcal{N}_{x_i}$. In this paper we consider the following two weighting schemes:

- Uniform: A baseline weighting scheme that assigns a uniform weight to all neighborhoods, i.e., $e(\mathcal{N}_{x_i}) = 1$ for $\forall\, \mathcal{N}_{x_i} \in \mathcal{N}$, which implies we assume all neighbor sets are equally likely to be selected.
- Connectivity: A node connectivity based weighting scheme that assigns higher weights to the neighbor sets that are located in "dense" areas of the graph $W$, i.e. $e(\mathcal{N}_{x_i}) = \sum_{j=1}^{n} W_{ji}$. This implies we prefer to query neighbor sets that are highly connected to other nodes both within and outside the neighborhood since they are more influential in label propagation.

After the most informative neighborhood set is selected, a non-expert will be asked to order the neighbors with respect to the similarity to the instance they are the neighborhood set of. It is possible that sometimes that a non-expert may not be able to confidently provide a complete ordering on neighbors, in this case only a partial ordering is acquired and this still helps the learning of graph weights. An example query in Galaxy Zoo data is shown in Figure 3. Though non-experts may not label the query Galaxy, they can provide an ordering of its neighbors based on their visual similarity to the query Galaxy. We can then see that our approach can benefit from people's visual perception being able to better organize the neighborhood than can be calculated from the data.

### 3.4 Implementation

In order to reduce the human efforts of ordering a neighbor set as well as to avoid excessive label propagation, the number of neighbors of each instance needs to be limited. In our implementation, we discard the neighbors whose weights in $W_{i\bullet}$ are under a certain threshold ($0.01$ in our experiment). The QP described in Eq.(2) and the LP relaxation of Eq.(6) are standard problems, thereby can be solved using any QP or LP solvers. In our experiments we use the built-in QP and LP solvers of Matlab. Our method can be easily scaled in a number of ways. Firstly, one may employ kd-trees [Panigrahy, 2008] or locality sensitive hashing [Gionis *et al.*, 1999] to efficiently construct the neighbor sets $\mathcal{N}$. Secondly, the learning of graph weights defined in Eq.(2) could be parallelized, since the weights for reconstructing each instance are solved independently of others. Thirdly, the label propagation defined in Eq.(3) also can be parallelized since matrix multiplication can be parallelized. Finally, to accelerate the selection of neighbor sets, one may use more efficient counting algorithms such as compressed-IC [Gionis *et al.*, 2012].

## 4 Empirical Evaluation

In this section we attempt to understand the strengths and relative performance of our approach `ALNO`. In particular we wish to answer how well our method compares to:

1. `Random+Label`: A baseline approach of random instance selection for querying labels.
2. `Active Harmonic Function`: A state-of-the-art active label propagation algorithm [Zhu *et al.*, 2003b; 2003a]. Its query strategy is expected risk minimization.

The surprising answer is that `ALNO` performs comparably as typical active learning despite not adding more labels, instead just improving neighborhood structure. Given this, an interesting question is, "Is the good performance due to the query selection strategy or the neighborhood ordering technique?". To investigate this we explore the following scenarios:

3. `Active+Label`: Our active query strategy with connectivity weighting scheme (Section 3.3) but querying labels rather than neighborhood structure (ask for the label of $x_i$ rather than ordering of $\mathcal{N}_{x_i}$).
4. `Random+Neighbor`: A random query selection but using our approach of querying neighborhood structure.

We compare these six methods, including the two versions of ALNO (two weighting schemes as shown in Section 3.3), on real datasets. In the first application *Galaxy Zoo* the queries are answered by crowd-sourcing whilst in the second

*Alzheimers Prediction* the queries are answered by side information (the patient's personal information). The parameters used in these algorithms ($\sigma$ in Active Harmonic Function and $\mu$ in ALNO) are optimized using cross-validation. To reduce the number of neighbors, for ALNO we discard the neighbors with a weight less than $0.01$. As the question queried in our model is the neighbor ordering rather than label, theoretically, it is possible (unlikely in practice) that the orderings provided by non-experts do not change the graph weights thus the learning model gains no useful information. To evaluate this, we define a measure called *hit-rate*, which refers to the fraction of times that the provided neighbor orderings changed the graph weights. The *hit-rate* implies the amount of successful knowledge injection of querying neighborhood structure. Specifically, a low *hit-rate* implies querying the neighbor structure did not introduce much new information to the learning model, while a $100\%$ *hit-rate* indicates that every ordering queried did change the learning model.

### 4.1 Application 1 – Galaxy Zoo

**Dataset and Experiment Settings.** In the Galaxy Zoo 1 project [Lintott *et al.*, 2011], volunteers are asked to annotate approximately $900,000$ galaxies into several categories, such as spiral, elliptical, uncertain. This is a very difficult task for people without an astronomy background and there is considerable noise throughout the data not only due to the limited resolution or errors of telescope, but also due to the labeling task itself. However, it would be much easier if we ask for the ordering of neighbors of a galaxy image, and most people can answer this kind of question correctly without knowing astronomy. In the experiment, we work on a subset of the Galaxy Zoo data, which consists of the first $3,000$ images of Galaxy Zoo 1 data release. The raw images were crawled from the Sloan Digital Sky Survey, and then preprocessed using a 2-D discrete cosine transform (DCT). After a zigzag scan on the transformed images, we take the top $100$ DCT coefficients to represent each original image. To allow reproducibility, the neighbor orderings are generated using the voting information provided in the data (real crowd-sourced values from non-expert volunteers). As the voting reveals people's visual perception of the galaxies, we believe it is a good analogy to the orderings that will be provided by human.

**Results and Discussion.** In each trial, we randomly select half of the galaxy images as the training set, and the rest were used for testing. The experiments are repeated for $30$ times, and the mean error rates are reported in Figure 4. We see that both the random methods `Random+Label` and `Random+Neighbor` are not helpful and even destructive to the learning accuracy. This confirms the motivation and necessity of active learning since asking randomly selected questions may not effectively improve the learning performance. Among the four active query methods, the best accuracy is achieved by `Active Harmonic Function` and `Active+Label`, which means that querying labels is preferable than querying neighborhood structure for this data set. However, it can be seen that our `ALNO-Connectivity` model achieves comparable learning accuracy with the two models without adding more labels, which confirms that our motivation that asking the easier question (neighbor ordering) can also efficiently improve the learning performance. For the two weighting schemes of `ALNO`, connectivity significantly outperforms uniform as the graph structure is factored into the selection of queries. Furthermore, the *hit-rate* of both the `ALNO` variations are $100\%$ which is higher than the $80.33\%$ *hit-rate* of `Random+Neighbor`. Since the randomly selected neighbor sets are not noticeably helpful when relearning $\mathbf{W}$, the usefulness of our neighborhood selection strategy is validated.

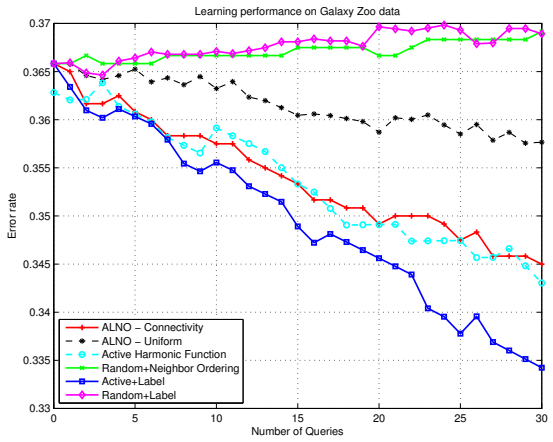
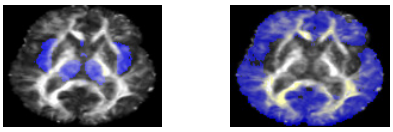



Figure 4: Error rate on Galaxy Zoo data

### 4.2 Application 2 – Prediction of Alzheimer's

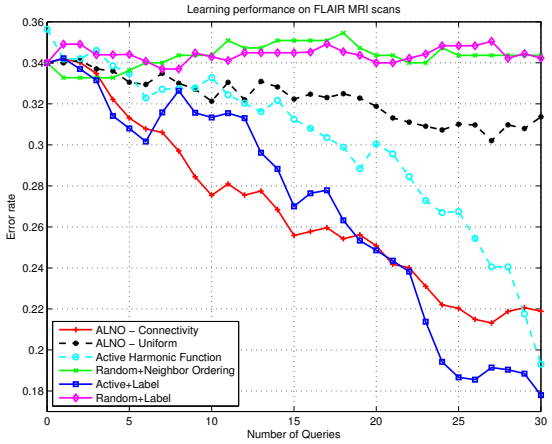


(a) White matter (b) Gray matter

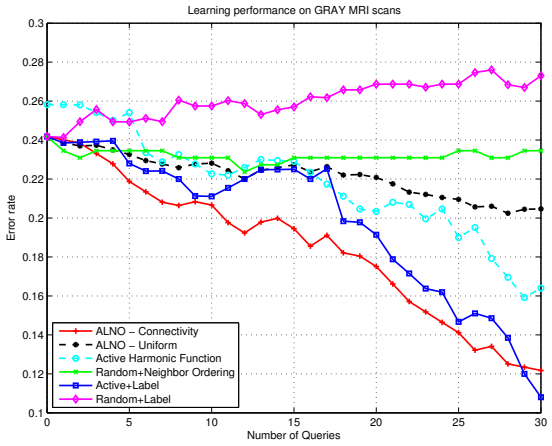
Figure 5: Example slice of structural MRI scan

**Dataset and Experiment Settings.** Structural MRI scans were acquired from real clinic cases of 632 patients, which is a *new dataset* and will be made publicly available. There are two types of MRI scans that were collected: (1) **FLAIR**: Fluid attenuated inversion recovery is a pulse sequence used in MRI, which carries the white matter hyper-intensity of a brain; (2) **GRAY**: T1-weighted MRI images which only reveals structural integrity of the gray matter of a brain. In the raw scans, each voxel has a value from 0 to 1, where 1 indicates that the structural integrity of the neuron cell bodies at that location is perfect, while 0 implies either there are no neuron cell bodies or they are not working. An example of the two types of scans is shown in Figure 5. The raw scans are preprocessed (including normalization, denoising and alignment) and then restructured to 3D matrices with a size of $134 \times 102 \times 134$. The learning problem is to determine if a person is normal, mildly cognitively impaired (MCI), or has dementia based on his or her brain structural MRI scan. Experienced clinicians or doctors may answer this question

correctly, but for most people it is impossible to judge a person's mental health condition using MRI scans. However, by careful visual comparison of the similarities and differences between these scans, many people could provide at least a partially correct neighbor ordering. To allow reproducibility, the role of the non-expert is played here by side information, where only an approximately correct neighborhood ordering is generated using the patient's personal information, including age, gender, race, and education. The collected personal information is a weaker predictor of the mental health condition, and therefore this composes a fair comparison, as neighborhood ordering is also a weak predictor of the label.




(a) Error rate on FLAIR scans




(b) Error rate on GRAY scans

Figure 6: Performance comparison on structural MRI scans

**Results and Discussion.** The experiment is repeated for 30 times. In each trial we randomly select $50\%$ of the MRI scans for the training, and perform testing on the remaining scans. The mean error rates are reported in Figure 6, where Figure 6(a) shows the result on FLAIR MRI scans and Figure 6(b) shows the result on GRAY scans. We see that the performance of the two random querying methods is very weak as they do not improve the learning accuracy and sometimes are even destructive. From the results we can observe that the connectivity is definitely a better weighting for our ALNO approach. It also can be seen that in this application there are two methods using our query strategy, `ALNO-connectivity` and `Active+Label`, outperform `Active Harmonic Function`, which demonstrates the effectiveness of our counting set cover strategy. Surprisingly, the performance of `ALNO-connectivity` is not just comparable to the label querying methods, but sometime even performs better. A plausible explanation is that MRI scans are complicated objects and lie in very high dimensional space, and therefore for a learning model it is difficult to understand the objects and construct the correct neighborhood structure directly from the features. Hence, in this application providing a few labels may not improve the learning model much, but providing a few key neighbor orderings could enhance the graph structure significantly and better propagate those labels already given. This illustrates the benefits of neighborhood structure querying, and implies that our `ALNO` model is more suitable to the learning problems involving complicated objects or high dimensional data. Moreover, for FLAIR scans the *hit-rate* of our query strategy is $100\%$, while `Random+Neighbor` only reaches a *hit-rate* of $72.67\%$; for GRAY scans the *hit-rate* of ours is $100\%$, while random querying being only $83.33\%$. We see that though, similar to our `ALNO` model, `Random+Neighbor` also modifies the learning model by changing the graph weights, those changes are not helpful to the learning performance. This implies that random querying may not help the learning, and demonstrates the demand for the proposed query strategy.

## 5 Conclusion

In this paper we present an alternative to label focused active learning and describe an approach that queries the neighborhood ordering of an instance. The proposed relative queries take the form of, "Is instance $i$ more similar to instance $j$ than instance $k$?" and can be easily answered by non-experts or even generated using side information. Our ALNO approach is easy to implement and higher efficiency can be obtained using parallelization. The promising experimental results demonstrate the usefulness of our approach as the neighborhood structure querying can achieve comparable and in some cases even better learning performance than label querying. It is important to note that the experiments were designed so that not only were the questions designed for non-experts, but the answers were provided by non-experts, e.g., in Galaxy Zoo experiment the answers were given by crowd-sourcing and for our MRI data from basic socio-demographic data. This is significant since it illustrates that non-expert advice cannot only be encoded but is useful when available.

## Acknowledgments

The authors gratefully acknowledge support of this research from ONR grants N00014-09-1-0712, N00014-11-1-0108 and NSF Grant NSF IIS-0801528. The authors thank Professor Owen Carmichael from Department of Neurology at UC Davis and UC Davis Alzheimers Disease Center for providing valuable comments and the MRI data, which is supported by NIH grants P30 AG010129 and K01 AG030514.

## References


[Chattopadhyay *et al.*, 2012] Rita Chattopadhyay, Zheng Wang, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Batch mode active sampling based on marginal probability distribution matching. In *KDD*, pages 741–749, 2012.

[Culotta and McCallum, 2005] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, pages 746–751, 2005.

[Gionis *et al.*, 1999] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *VLDB*, pages 518–529, 1999.

[Gionis *et al.*, 2012] Aristides Gionis, Theodoros Lappas, and Evimaria Terzi. Estimating entity importance via counting set covers. In *KDD*, pages 687–695, 2012.

[Guo and Greiner, 2007] Yuhong Guo and Russ Greiner. Optimistic active learning using mutual information. In *IJCAI*, pages 823–829, 2007.

[Hoi *et al.*, 2006] Steven C. H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. Batch mode active learning and its application to medical image classification. In *ICML*, pages 417–424, 2006.

[Kapoor *et al.*, 2007] Ashish Kapoor, Eric Horvitz, and Sumit Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI*, pages 877–882, 2007.

[Lewis and Gale, 1994] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *SIGIR*, pages 3–12, 1994.

[Lintott *et al.*, 2011] C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R. C. Nichol, M. J. Raddick, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. Galaxy Zoo 1: data release of morphological classifications for nearly 900 000 galaxies. *Monthly Notices of the Royal Astronomical Society*, 410:166–178, January 2011.

[Melville and Mooney, 2004] Prem Melville and Raymond J. Mooney. Diverse ensembles for active learning. In *ICML*, pages 74–81, 2004.

[Muslea *et al.*, 2000] I. Muslea, S. Minton, and C. Knoblock. Selective sampling with redundant views. In *Proceedings of the National Conference on Artificial Intelligence*, 2000.

[Panigrahy, 2008] Rina Panigrahy. An improved algorithm finding nearest neighbor using kd-trees. In *LATIN*, pages 387–398, 2008.

[Rashidi and Cook, 2011] Parisa Rashidi and Diane J. Cook. Ask me better questions: active learning queries based on rule induction. In *KDD*, pages 904–912, 2011.

[Roweis and Saul, 2000] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[Roy and Mccallum, 2001] Nicholas Roy and Andrew Mccallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, pages 441–448, 2001.

[Settles and Craven, 2008] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP*, pages 1070–1079, 2008.

[Settles *et al.*, 2008] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *NIPS*, pages 1289–1296, 2008.

[Settles, 2009] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[Tamuz *et al.*, 2011] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Kalai. Adaptively learning the crowd kernel. In *ICML*, pages 673–680, 2011.

[Vazirani, 2001] Vijay V. Vazirani. *Approximation algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[Wang and Zhang, 2006] Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. In *ICML*, pages 985–992, 2006.

[Wauthier *et al.*, 2012] Fabian L. Wauthier, Nebojsa Jojic, and Michael I. Jordan. Active spectral clustering via iterative uncertainty reduction. In *KDD*, pages 1339–1347, 2012.

[Zhang and Oles, 2000] Tong Zhang and Frank J. Oles. A probability analysis on the value of unlabeled data for classification problems. In *ICML*, pages 1191–1198, 2000.

[Zhou *et al.*, 2003] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.

[Zhu *et al.*, 2003a] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.

[Zhu *et al.*, 2003b] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65, 2003.