
Repairing Self-Confident Active-Transductive Learners Using Systematic Exploration

Ron Begleiter

Department of Computer Science
Technion
Haifa

ronbeg@cs.technion.ac.il

Ran El-Yaniv

Department of Computer Science
Technion
Haifa

rani@cs.technion.ac.il

Dmitry Pechyony

Department of Computer Science
Technion
Haifa

pechyony@cs.technion.ac.il

Abstract

We consider an active learning game within a transductive learning model. A major problem with many active learning algorithms is that an unreliable current hypothesis can mislead the querying component to query “uninformative” points. In this work we propose a remedy to this problem. Our solution can be viewed as a “patch” for fixing this deficiency and also as a proposed modular approach for active-transductive learning that produces powerful new algorithms. Extensive experiments on “real” data demonstrate the advantage of our method.

1 Introduction

Efficient utilization of unlabeled examples during a learning process can be very advantageous when constructing accurate classifiers using limited labeling efforts. Prominent approaches for achieving this goal are semi-supervised, transductive, and active learning. In this paper we focus on active classification problems within a transductive setting. Given a sample of unlabeled examples and a *labeling budget* m , the learner should select m examples to be labeled by the teacher. The goal is to use the m labeled examples to classify the rest of the points in the sample. This model is similar to *inductive* pool-based active learning but, in the transductive setting, the learner is tested only over the given (remaining) pool points. The learner can influence the choice of test points (e.g., by not asking for labels of “easy” points). In addition, the size of the hypothesis class from which the learner selects the final classifier is effectively finite, which may ease the learning process.

Despite the attractiveness of this *active-transductive learning* setting, most of the research contributions on active learning have focused on inductive models. A few studies do consider the above model [1, 2, 3]. These works tend to rely on graph-based algorithms, which have been used extensively in transductive settings [4]. Our research motivation is the observation that the known active-transductive algorithms (as well as many active-inductive algorithms) tend to suffer from excessive “self-confidence,” which can severely impair their performance. This flaw, which results in the neglect of whole areas of the input space during early stages of the learning process, is demonstrated in Section 1.1. We propose a simple yet effective solution that enforces systematic exploration of the input space whenever it is necessary. Our `+EXPLORE` method can be viewed as a “patch” for fixing this deficiency and also as a proposed modular approach for generating effective new active-transductive algorithms that clearly beat currently available active-transductive algorithms.

A few previous works consider the above deficiency of self-confident active-inductive learning algorithms. All their solutions are based on ensemble methods. In [5], it was demonstrated that self-confident learners fail on XOR-like problems. Their solution provides a general framework for combining a set of active learning schemes. The framework is based on online learning algorithms for the multi-armed bandit problem. A simpler solution by [6] more directly relates the deficiency to the classical exploration-exploitation problem. They switch between a self-confident learner and an “exploration” learner whenever the induced hypothesis does not change “much.” The ensemble methods employed in [5, 6] depend on a number of hyper-parameters and there is no clear way to calibrate them. A very recent simpler idea by [7] combines two active learners using a round-robin-like scheme; however, the learners employed in this solution are both self-confident. Thus, the combined algorithm tends to suffer from the above deficiency.

Our proposed +EXPLORE solution is based on cluster covering. Other works considered clustering within the context of active (inductive) learning. The active method of [8] queries cluster centers of the instances that lie within the margin of the support vector machine. The algorithm of [9] combines clustering and active learning. However, their method is not general purpose and the switch between the baseline active learner and the clustering depends on a predefined hyper-parameter.

1.1 Motivating Example, and a Preview

The active-transductive algorithm of Zhu et al. [1] is among the few known algorithms designed for the active-transductive game (more details on this algorithm appear in Section 4.1). The starting point of the current paper was our empirical evaluation of this algorithm (as well as others), which showed that it is a top performer in this setting. Our initial study also revealed a major deficiency, depicted in Figure 1. The synthetic example in Figure 1(i) is a binary classification problem with three (non-isotropic) Gaussians and two “outlier” points that reside between the lower Gaussians. Applied on this example¹, algorithm [1] exhibits behavior that is summarized by the upper learning curve of Figure 1(ii); it does not query any point within the lowest Gaussian and fails to decrease its error below 15%, within the first 100 active queries. Of course, this example was carefully constructed to emphasize this bad behavior, but it does represent a reoccurring behavior we observed on many “real” datasets. The +EXPLORE method we develop here salvages this algorithm w.r.t. such learning problems without reducing its already good performance on easier problems.

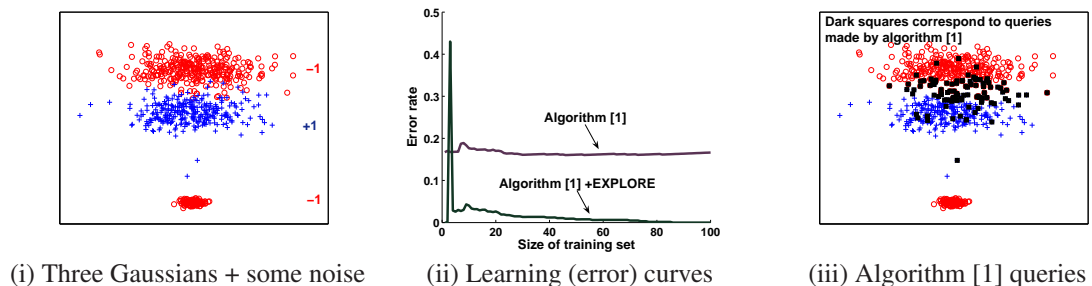


Figure 1: Motivating example

2 Problem Definition

The distribution-free transductive setting [10, Chapter 10] is defined as follows. Consider a fixed set $S_{m+u} \triangleq \{(\mathbf{x}_i, y_i)\}_{i=1}^{m+u} \subseteq \mathbb{R}^d \times \{+1, -1\}$ of $m + u$ points along with their *binary* labels. The learner is provided with the unlabeled *full-sample* $X_{m+u} \triangleq \{\mathbf{x}_i\}_{i=1}^{m+u}$. A *training set* S_m consisting of m labeled points (\mathbf{x}_i, y_i) is selected from S_{m+u} uniformly at random among all subsets of size m and is given to the learner. The *test set* X_u of size u , containing the remaining *unlabeled* points, is also given to the learner. The learners we consider here generate *soft classification* vectors $\mathbf{h} \triangleq$

¹We set the hyper-parameter $k = 10$ in the algorithm of [1], and noticed that the algorithm failed with other k values on similar examples. The noise is essential for establishing this example, because this algorithm does handle noiseless XOR-like problems (see, e.g., [1, Figure 2]).

$(h_1, \dots, h_{m+u}) \in \mathbb{R}^{m+u}$, where h_i is the soft label of example \mathbf{x}_i given by the *hypothesis* \mathbf{h} . The algorithm outputs $\text{sign}(h_i)$ for the actual (binary) classification of \mathbf{x}_i . In the *passive* transductive setting the goal of the learner is to predict the labels of the test points in X_u from (S_m, X_u) , so as to minimize the transductive risk, $\frac{1}{u} \sum_{\mathbf{x}_i \in X_u} \ell(\text{sign}(h_i), y_i)$, w.r.t. the 0/1 loss function ℓ . In this paper our focus is on *active* transductive learning. In this setting the m training points are actively selected by the learner. The examples to be queried are selected iteratively. At each iteration the learner selects the next example to be queried and receives its label from a teacher.² The goal of the active learner is to minimize the transductive risk over the remaining points that were not queried.³

3 An Exploration-Exploitation Routine and Its Implementation

Our starting point is an active learning algorithm $\text{ALG} = (P, Q)$ where P is a passive learning algorithm and Q is a querying component. The passive learner P uses a given (S_m, X_u) to generate a transductive hypothesis \mathbf{h} and the querying component selects the next example to query $\mathbf{x} \in X_u$ using \mathbf{h} and (S_m, X_u) . In this section we describe the `+EXPLORE` routine, whose goal is to improve the performance of such algorithms by enforcing systematic exploration of unlabeled points. The proposed routine requires two components: an auxiliary querying function $Q_A(S_m, X_u)$ and a switching function $\text{SW}(S_m, X_u)$ that determines whether to generate the query using Q or Q_A . Note that Q_A and SW do not rely on \mathbf{h} and thus do not suffer from the “self-confidence” deficiency.

Our routine can be viewed as a meta-algorithm that operates the given active learner and augments it with additional querying capabilities. The routine performs m iterations corresponding to the m required queries. At each iteration, the switching component defines which querying method to apply next. Upon termination, the routine uses the passive learner P and the aggregated training set S_m to classify the remaining test points.

The pseudocode in Figure 2 defines our proposed procedure. In the following sections we describe our implementation of the decision function SW and auxiliary querying component Q_A (Section 3.1). In our experiments we examined several implementations of the Q and P components; these are described in Sections 3.2 and 4.1.

3.1 Implementation of Q_A and SW

Our implementation of Q_A and SW is based on a very simple and effective method of cluster covering. At each iteration we cluster X_u . If there is an uncovered cluster containing no labeled points, our switching function SW decides to use Q_A , which selects a “representative” point from the largest uncovered cluster. Otherwise, SW operates the original querying function Q .

The clustering we perform in this implementation is *semi-supervised*, which means that it takes into account all available points and labels. Since the set of acquired labels grows during the active learning session, the clustering we compute is dynamically improved after each iteration.

In the rest of this section we describe our implementation of the semi-supervised clustering. At the i th iteration we build a graph G_i representing the current training and test sets $(S_{i-1}, X_{m+u-i+1})$. This is done in two steps. In the first step we generate a symmetric kNN graph, denoted by G , which

²There is also a “batch” form of selection of training examples, in which the learner selects the examples for the training set and then simultaneously queries all of them (e.g., see [3]). Batch querying is a special case of sequential querying and can be viewed as a limitation of the learner.

³Alternate optimization criteria may consider the entire learning curve of the active learner (e.g., see [5, 11]).

Input: The unlabeled full sample X_{m+u} ; an active learning algorithm $\text{ALG} = (P, Q)$; a switching component SW ; and an auxiliary querying component Q_A .
Output: A classification of X_{m+u} .
 $S_0 = \emptyset$.
for $i = 1$ to m **do**
 $\mathbf{h} = P(S_{i-1}, X_{m+u-i+1})$.
 if $\text{SW}(S_{i-1}, X_{m+u-i+1}) == Q$ **then**
 $\mathbf{x} = Q(\mathbf{h}, S_{i-1}, X_{m+u-i+1})$.
 else
 $\mathbf{x} = Q_A(S_{i-1}, X_{m+u-i+1})$.
 end if
 Query the label y of the \mathbf{x} point.
 $S_i = S_{i-1} \cup \{(\mathbf{x}, y)\}$.
 $X_{m+u-i} = X_{m+u-i+1} \setminus \{\mathbf{x}\}$.
end for
return $P(S_m, X_u)$.

Figure 2: `+EXPLORE` routine

represents the unlabeled full sample X_{m+u} . In this graph there is an edge between two points iff one of them is among the k “most similar” points to the other. We measure the similarity between \mathbf{x}_i and \mathbf{x}_j by the cosine similarity, $\mathbf{d}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2 / (|\mathbf{x}_1| |\mathbf{x}_2|)$. We note that this choice of metric is arbitrary and any metric could be used. If there exists an edge between the points \mathbf{x}_i and \mathbf{x}_j then we set its weight to be $w_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$; otherwise, $w_{ij} = 0$.

Starting with G , we then construct the graph G_i , which encodes all known labels (in S_{i-1}). Many methods have been proposed for incorporating labeled points in clustering (see [12] and the references therein). We tried several of them and obtained rather weak results in our setting. Hence, we propose a novel heuristic, which is guided by the following commonly used principles [12]:

1. Points with different labels should not in general be “similar.” Thus, we delete the edges between such points in S_{i-1} (by setting their weights to zero).
2. Points with the same label can be similar. Hence, if there exists a pair of points \mathbf{x}_r and \mathbf{x}_s in S_{i-1} with the same label and there is no edge w_{rs} between them, we add an edge whose weight is $w_{rs} = \frac{1}{2} (\min_{j:w_{rj} \neq 0} \{w_{rj}\} + \min_{j:w_{sj} \neq 0} \{w_{sj}\})$.

After G_i is constructed, we cluster it using a graph-based (pairwise) clustering algorithm. In general, any unsupervised clustering algorithm can be used. We preferred an algorithm that has some kind of “principled” mechanism for selecting the number of clusters.⁴ In light of our familiarity and extensive experience with spectral techniques (such as those discussed in [13, 14]), we used the *Eigenvector-Alignment* mechanism of [15], which is summarized in Appendix A.

It remains to describe our implementation of \mathcal{Q}_A , the auxiliary querying component. As mentioned above, given the largest uncovered cluster, our goal is to select a representative point in this cluster. A representative point can be defined in several meaningful ways. For example, it can be referred to as the most central point in the cluster (in the sense of minimizing the maximal distance to any point). While this approach makes sense, it seems to be computationally expensive (for example, it takes cubic time in cluster size using the Floyd-Warshall algorithm to select a representative point). Therefore, we defined the representative point as the one that is most similar to its neighbors, namely, the point with the largest sum of weights of its edges. This point can be identified in quadratic time in the cluster size.

3.2 On Some Known and Some New Querying Components

In this section we describe several known and some new querying components \mathcal{Q} that were used in our experiments. The first active querying method that we consider is a transductive variant of the *worst case* heuristic of [16]. This heuristic is motivated by the following considerations. We assume that the absolute value $|h_i|$ of the soft-classification of the i th point is proportional to the true probability that its label y_i is $\text{sign}(h_i)$, and choose to query $\mathbf{x} = \text{argmax}_{\mathbf{x}_i \in X_u} \min\{(1 - h_i)^2, (-1 - h_i)^2\}$. It can be verified⁵ that the values of \mathbf{h} produced by the passive algorithms that we consider (see Section 4.1) are in $[-1, 1]^{m+u}$. Therefore, the solution for the above optimization problem is the most uncertain point, $\text{argmin}_{\mathbf{x}_i \in X_u} |h_i|$. We term this method “UNCERTAIN.” Some active-transductive experiments with the UNCERTAIN querying function are presented in [1, 2].

When operated with Support Vector Machines (SVM), UNCERTAIN coincides with the minimum margin method (called SIMPLE in [17]), which queries the point with the minimal distance to the separating hyperplane. We propose a transductive variant of the SIMPLE strategy. A graph cut between positive and negative vertices, induced by \mathbf{h} , can be considered as a transductive variant of the separating hyperplane. Hence, the transductive analogue, denoted by CUT, queries the unlabeled point that is closest to the cut. The distance to the cut is measured according to the edge weights, and the larger the path to the cut is, the closer the point is. The ties are resolved by a random selection among the closest points to the cut. Unlike SVM, in graph-based algorithms the UNCERTAIN and CUT methods can query different points, although they mostly query points lying on the graph cut (i.e., the points \mathbf{x}_i such that there exists $\mathbf{x}_j, w_{ij} \neq 0$ and $\text{sign}(h_i) \neq \text{sign}(h_j)$).

⁴While we do recognize that an automatic selection of the number of clusters is generally an ill-defined problem, some algorithms have reasonable justifications for offering selection mechanisms.

⁵If the absolute values of some set of h_i s exceed ± 1 , then cutting all of them to $\text{sign}(h_i)$ will only reduce the training error and the regularization term.

Recall that the active-transductive setting differs from the pool-based setting in that the learner is tested on the pool’s unlabeled points. Thus, a possible querying strategy could be to remove “hard” points from the pool. We define this “hardness” according to the difference between the (soft) classifications of a point and its neighbors. For example, the hardness of \mathbf{x}_i may be defined to be $\sum_{j=1}^{m+u} (h_i - h_j)^2 w_{ij}$. Intuitively, such points reside in regions which include many close points with opposite labels. We denote by COARSE this novel method that queries the most coarse point.

The last two active querying methods we examined are those of Zhu et al. [1] and Herbster et al. [2]. The method of [1], termed here REDUCE-RISK, queries the point that minimizes the expected transductive risk of the underlying passive algorithm. Since the true risk cannot be computed, [1] approximated it by the overall uncertainty over the test set, $\sum_{\mathbf{x}_i \in X_u} |h_i|$. The naïve implementation of REDUCE-RISK is computationally intensive, since for each query it needs to run the passive classifier $\Omega(u)$ times. They developed an efficient implementation for their passive algorithm [18].

The querying function of [2] queries the point that optimizes the trade-off between being uncertain and being central (namely, the distance from it to any point in the graph). This heuristic is motivated by the bound on the number of mistakes made by the underlying online algorithm of [2].

4 Empirical Evaluation

We empirically validated the efficiency of the +EXPLORE procedure using 14 different self-confident active-transductive algorithms and 11 standard datasets. Among these algorithms, two are known [1, 2] and the rest are novel.

4.1 Review of Graph-Based Transductive Algorithms Used in Our Experiments

We focus on four graph-based transductive algorithms⁶ by [20, 18, 21, 22]. These algorithms generate smooth solutions, namely the soft-classification does not change much between nearby points.

Let $\mathbf{y} \in \{-1, 1, 0\}^{m+u}$ be a vector of known labels defined as follows: if $\mathbf{x}_i \in X_u$, then the i th entry in \mathbf{y} is 0; otherwise, the i th entry in \mathbf{y} is y_i (the true label of \mathbf{x}_i). All four algorithms minimize the objective function $\min_{\mathbf{h} \in \mathbb{R}^{m+u}} \mathbf{h}^t \mathbf{R} \mathbf{h} + c(\mathbf{h} - \mathbf{y})^t \mathbf{C}(\mathbf{h} - \mathbf{y})$, where the left-hand term is a regularization term corresponding to the smoothness requirement and the right-hand term corresponds to the loss of the hypothesis \mathbf{h} . The constant $c \in \mathbb{R}$ provides a balance between the regularization and loss terms.

The *regularizer* \mathbf{R} is an $(m + u) \times (m + u)$ matrix induced by an adjacency matrix \mathbf{W} . The adjacency matrix reflects the similarity between the full-sample points. In our experiments we used the adjacency matrix corresponding to the kNN graph \mathbf{G} , which is built as described in Section 3.1. All four algorithms use the graph Laplacian regularizer $\mathbf{L} \triangleq \mathbf{D} - \mathbf{W}$, or its normalized version $\mathbf{L}_{norm} \triangleq \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}$, where \mathbf{D} is a diagonal matrix with the (i, i) th entry $d_i = \sum_{j=1}^{m+u} w_{ij}$ and \mathbf{I} is an identity matrix. The *cost* matrix \mathbf{C} is an $(m + u) \times (m + u)$ diagonal matrix with the (i, i) th entry being a misclassification cost for the i th example. All examples in the training (test) set have the same misclassification cost, denoted by C_m (C_u).

The Gaussian random field model (GRFM) algorithm [18] sets $\mathbf{R} = \mathbf{L}$, $C_l = \infty$ and $C_u = 0$. Hence this algorithm generates the solution with zero training errors. The algorithm of [21] also sets $\mathbf{R} = \mathbf{L}$ and $C_u = 0$. However, in this algorithm $C_l = 1$ and hence empirical errors are allowed.⁷ We refer to this algorithm as SOFT. The Spectral Graph Transducer (SGT) algorithm [20] sets $\mathbf{R} = \mathbf{L}$, $0 < C_l < \infty$, and $C_u = 0$. However, SGT adds two constraints: $\sum_i h_i = 0$; and $\sum_i h_i^2 = m + u$, imposing solutions that minimize the *ratiocut*, induced in \mathbf{G} by positive and negative values in \mathbf{h} . The consistency method (CM) [22] sets $\mathbf{R} = \mathbf{L}_{norm}$ and $C_l = C_u = 1$. This value of C_u forces the soft classification of the unlabeled points that are far from the labeled ones to be close to zero.

⁶The application of our active learning scheme to other transductive algorithms is straightforward. See [19] for a comprehensive survey of the existing transductive algorithms.

⁷In addition, this algorithm uses the constraint $\sum_i h_i = 0$, which is required for proving a risk bound.

4.2 Datasets and Experimental Setting

The comparison is made on 11 datasets: PIMA, BUPA, VOTING, TAE, IONOSPHERE, MUSH, MUSK, MONK, COIL, DIGIT, and TEXT. These datasets are used in the context of empirical validation of transductive algorithms. The first eight datasets are standard UCI datasets used by [23];⁸ the image datasets COIL and DIGIT are used by [4]; and the 20-newsgroups’ binary sub-problem “Atheism -vs- Religion” TEXT was used by [1]. All datasets were shuffled and cut in half.⁹

We ran GRFM, SOFT, SGT, and CM passive learners with the following querying components: CUT, UNCERTAIN, and COARSE. In addition, we experimented with the active-transductive algorithms of [1] and [2]. We term all these (P, Q) -combination algorithms as SELF-CONF (P, Q) .

Recall that SELF-CONF (P, Q) algorithms base their query on a transductive hypothesis and thus require an initial training set consisting of two examples. Hence, we report the mean error of such learners over five initialization chosen uniformly at random. Note that our +EXPLORE procedure always starts with exploration steps and thus implies a deterministic choice of the initial training set.

We report the best result in hindsight achieved over a grid of hyper-parameters. In general, no parameter selection scheme exists for (both transductive and inductive) active learning. Hence the goal of this section is to explore the potential of +EXPLORE on top of SELF-CONF. The grid of $k \in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 75, 100\}$ was shared by all of the algorithms. The adjacency matrices \mathbf{W} were built with cosine similarity. We used the following values for c , the hyper-parameter that balances the regularization and loss terms : $\{0.001, 0.01, 0.1, 1, 10, 100\}$ in SOFT and CM; $\{0.1, 1, 10, 100, 1000, 3200\}$ in SGT¹⁰.

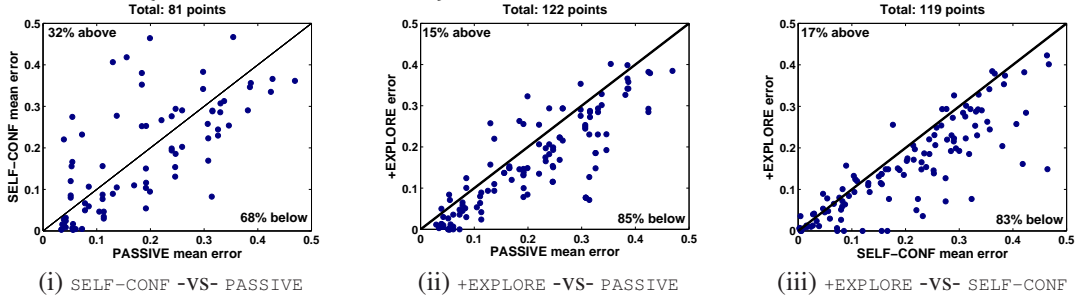


Figure 3: Comparing the three methods: PASSIVE, SELF-CONF, and +EXPLORE. Each point in each axis comprises two mean error results of two methods (in the x-axis and the y-axis) over one of the datasets for a training size $m = 50$.

4.3 The Efficiency of +EXPLORE

Figure 3(i)-(iii) depicts scatter plots comparing the +EXPLORE, SELF-CONF, and PASSIVE (P) methods. The comparison comprises 154 experiments and was carried out over all datasets, using all SELF-CONF (P, Q) combinations, and the known active-transductive algorithms of [1, 2]. Notice that most of these experiments correspond to new SELF-CONF (P, Q) combinations that have not been tested before. A point above (below) the dividing line corresponds to a “loss” (win) of the y-axis method over the x-axis. We depict only results for which there is no overlap between the corresponding mean error \pm the standard error of the mean (SEM).

Observe that the SELF-CONF methods are better than PASSIVE only over 55 out of 81 results. When applying SELF-CONF together with +EXPLORE, the advantage over PASSIVE is increased to 104 out of 122 results. Note that the number of significant wins over P is increased by 89% when using +EXPLORE. This effect is confirmed by Figure 3(iii), which depicts the clear advantage of +EXPLORE over SELF-CONF.

Next we compare in Table 1 the “best” representative of each method. These representatives were chosen according to the Friedman rank test [24] at 95% significance level. For completeness we also

⁸Some of these UCI datasets contain nominal features, which we translated into a vector of indicator bits.

⁹This was done to reduce the amount of running time, which took more than a month using 20 CPUs.

¹⁰All other hyper-parameters of the SGT implementation were set to their default values.

include the results of the relevant active-transductive algorithms of [1, 2]. The comparison shows that +EXPLORE achieves the best results on 9 out of the 11 datasets.

Data	P = SGT	SELF-CONF(P, Q)		SELF-CONF(P, Q)		Active-transductive algorithms of	
		(SGT, UNCERTAIN)	+EXPLORE	(CM, UNCERTAIN)	+EXPLORE	Zhu et al. [1]	Herbster et al. [2]
PIMA	29.8±0.4	31.1±1.0	27.5	28.8±0.5	27.2	28.9±0.8	29.0±0.0
BUPA	38.5±0.7	36.6±1.6	36.6	35.6±1.0	34.1	39.2±0.4	46.3±0.0
VOTING	5.6±1.0	0.6±0.5	0.6	0.5±0.2	0.0	1.2±0.0	4.6±0.22
TAE	30.8±2.7	22.3±2.8	7.7	15.4±2.1	11.5	20.0±1.5	36.2±0.9
IONOSPHERE	22.1±1.4	19.7±1.8	13.5	19.4±1.5	18.3	15.3±1.0	28.6±0.0
MUSH	6.1±1.8	0.4±0.0	3.6	0.6±0.4	0.0	3.3±0.6	8.0±0.3
MUSK	19.2±1.9	15.0±1.4	13.1	11.0±0.4	12.0	15.8±0.2	25.6±0.0
MONK	19.2±1.9	10.4±1.6	13.3	18.6±1.5	15.1	20.2±1.2	20.4±1.6
COIL	19.2±3.4	11.6±2.1	10.0	9.5±0.6	8.4	37.1±3.7	46.7±3.2
DIGIT	3.4±0.7	0.3±0.0	0.3	1.5±0.2	1.3	1.0±0.0	3.9±1.5
TEXT	7.8±0.7	5.0±0.3	4.5	10.5±0.5	9.4	11.1±0.2	13.3±0.8

Table 1: The error (%) of the “best” representatives of the +EXPLORE, Q, and P methods. The lowest error in each row appears in bold font.

4.4 The Advantage of Adaptive Exploration

We sketch a few numerical examples that indicate the usefulness of performing a dynamic exploration. This adaptive nature of exploration is crucial for establishing the advantage added by +EXPLORE to SELF-CONF algorithms. Figure 4(i) depicts how a bad choice of only three points at the beginning of the learning rounds dramatically affects the performance. The three dots on the error curve of +EXPLORE correspond to the exploration steps (determining the initial training set).

Figure 4(ii-iii) depicts the positive effect of performing adaptive exploration. Observe in Figure 4(ii) how the sequence of three exploration steps, starting around $m = 33$, separates the error curves of SELF-CONF and +EXPLORE. The sequence of explorations as depicted in Figure 4(iii) dramatically reduces the error rate from 0.2 to 0.

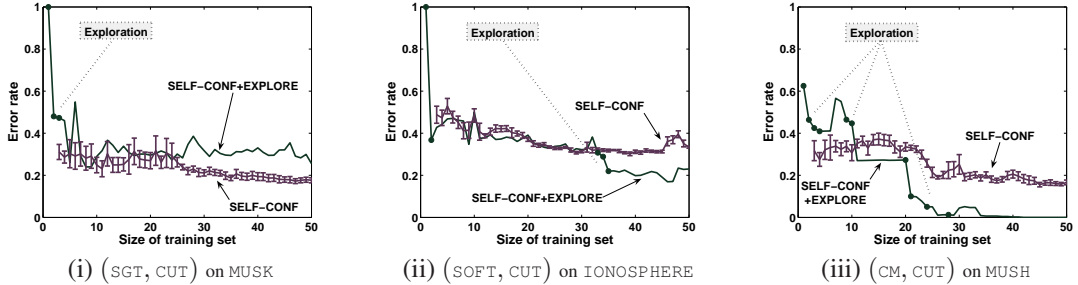


Figure 4: The effect of dynamic exploration: Comparing the learning (error) curves of SELF-CONF with SELF-CONF+EXPLORE. Queries by exploration (using Q_A) are indicated by dark dots.

5 Concluding Remarks

We propose a simple yet effective enhancement procedure that repairs the self-confidence deficiency of many active-transductive algorithms. We empirically tested our proposed +EXPLORE method using the known active-transductive algorithms of [1, 2] and 12 new algorithms. The experiments clearly indicate that our +EXPLORE enhancement improves the performance of self-confident active learners in most cases. Moreover, the state-of-the-art results are achieved when applying (SGT, UNCERTAIN) and (CM, UNCERTAIN) along with the +EXPLORE method.

Currently, our results rely on parameter selection in hindsight, which is the customary method in empirical evaluation of active learning algorithms. The reason is that no parameter selection scheme exists for (both transductive and inductive) active learning. Clearly, this is a major open problem.

While our experiments indicate the advantage of active over passive learning, the theoretical understanding of the active model is still in its early stages. Current theoretical works considered only

the active inductive setting. We believe that considering also the active-transductive setting may improve the current understanding. Finally, we note that our proposed solution defines a modular approach that is not limited to the active-transductive setting. We speculate that applying the same idea will be advantageous both in the *inductive* and *semi-supervised* settings.

References

- [1] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML workshop*, 2003.
- [2] M. Herbster, M. Pontil, and L. Wainer. Online learning over graphs. In *ICML*, 2005.
- [3] K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *ICML*, 2006.
- [4] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.
- [5] Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. *JMLR*, 5:255–291, 2004.
- [6] T. Osugi, D. Kun, and S. Scott. Balancing exploration and exploitation: A new algorithm for active machine learning. In *ICDM*, 2005.
- [7] Y. Guo and R. Greiner. Optimistic active-learning using mutual information. In *IJCAI*, 2007.
- [8] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang. Representative sampling for text classification using support vector machines. In *ECIR*, 2003.
- [9] H.T. Nguyen and A. Smeulders. Active learning using pre-clustering. In *ICML*, 2004.
- [10] V. N. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York, 1982.
- [11] P. Melville, M. Saar-Tsechansky, F. Provost, and R.J. Mooney. Economical active feature-value acquisition through expected utility estimation. In *KDD-05 Workshop on Utility-Based Data Mining*, 2005.
- [12] B. Kulis, S. Basu, I. Dhillon, and R.J. Mooney. Semi-supervised graph clustering: A kernel approach. In *ICML*, 2005.
- [13] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.
- [14] U. von Luxburg. A tutorial on spectral clustering. Technical Report TR-149, Max Planck Institute for Biological Cybernetics, 2007.
- [15] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *NIPS*, 2004.
- [16] C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *ICML*, 2000.
- [17] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *JMLR*, number 2, pages 45–66, 2001.
- [18] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.
- [19] X. Zhu. Semi-supervised learning literature survey. Technical Report TR-1530, University of Wisconsin-Madison, 2006.
- [20] T. Joachims. Transductive learning via spectral graph partitioning. In *ICML*, 2003.
- [21] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, 2004.
- [22] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2004.
- [23] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML*, 2001.
- [24] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30, 2006.

A The Zelnik-Manor and Perona (2004) Clustering Algorithm

Let k be a candidate number of clusters and $V = \{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ be the eigenvectors of the normalized Laplacian corresponding to its k largest eigenvalues. Let v_{ij} be the j th entry of \mathbf{v}_i . These k eigenvectors define a new representation of the full sample points: each $\mathbf{x}_j \in X_{m+u}$ is represented as $\tilde{\mathbf{x}}_j = (v_{1j}, v_{2j}, \dots, v_{kj})$. Let $E = \{\mathbf{e}_1, \dots, \mathbf{e}_k\}$ be a set of unit vectors such that the vector \mathbf{e}_i has 1 in the i th entry and 0 in other entries. The eigenvector-alignment mechanism rotates all vectors in V simultaneously. Each such rotation R induces a *rotated representation* $R(\tilde{\mathbf{x}}_j) \in \mathbb{R}^k$ of each $\tilde{\mathbf{x}}_j$. The idea of the eigenvector-alignment mechanism is to find the rotation R to maximally align each $R(\tilde{\mathbf{x}}_j)$, $1 \leq j \leq m + u$, with one of the vectors in E . In the “ideal” alignment, for

any $1 \leq j \leq m + u$ it holds that $R(\tilde{\mathbf{x}}_j) \in E$. In this case, each example \mathbf{x}_j is assigned to the cluster corresponding to the index of the nonzero coordinate of $R(\tilde{\mathbf{x}}_j)$. If the maximal alignment is not the “ideal” one, then each \mathbf{x}_j is assigned to the cluster corresponding to the maximal coordinate of $R(\tilde{\mathbf{x}}_j)$. The eigenvector-alignment mechanism chooses the value of k (among the list of the candidate values) providing the best alignment.