

# HIGCALs: a hierarchical graph-theoretic clustering active learning system

Wei Hu and Weiming Hu

**Abstract**—Active learning aims at automatically selecting highly informative unseen data for humans to label under the condition that only few labeled samples are ready for use. Most of previous researches in active learning take advantage of supervised learning. Due to semantic gap problem, these existing active learning algorithms are not very effective on lessening human labeling efforts, especially in multiclass applications. In this paper, we propose a novel active learning framework based on unsupervised learning, and implement a hierarchical graph-theoretic clustering active learning system (HIGCALs). HIGCALs outperforms the existing active learning systems in several aspects, such as flexibility in system architecture and simpleness in system upgrade. Experiments on *KDDCUP99* data set has demonstrated that HIGCALs can effectively reduce the workload of manual labeling without losing much accuracy.

## I. INTRODUCTION

In many applications such as image retrieval, video annotation and network intrusion detection, it is hard or prohibitive to get a great amount of labeled data. So we need a mechanism to automatically select highly informative data for manual labeling with relatively few labeled samples at hand. This is the very problem active learning aims at solving.

Most exiting active learning algorithms are based on supervised learning strategy [4], [2], [10]. Certainty-based methods use labeled data to train only one classifier. If a new sample falls close to the classification surface, it is considered as of low “certainty” thus high informative. Committee-based methods obtain several classifiers from labeled data. If a new sample arouse great “discrepancy” among these “committee” classifiers, it is considered highly informative [9].

But these active learning algorithms have some common problems. Supervised learning essentially tries to find a proper mapping  $f : \mathcal{X} \rightarrow \{1, \dots, L\}$  from feature space  $\mathcal{X}$  to a finite label set  $\{1, \dots, L\}$ . With few labeled samples and large semantic gap, the mapping trained by supervised learning algorithms will be very unstable, especially in multiclass applications. That makes the supervised-based active learning algorithms lack enough ability to estimate the information of a new sample thus cannot effectively reduce the human labeling efforts. Here we borrow the word “semantic gap” from computer vision, which means the feature space constructed by researchers is far from complete for capturing label information, that is to say, the relations between instances’ features and labels are weak to learn. In

computer vision, semantic gap shows up in the situation that the low level features such as color and shape cannot fairly reflect the high level semantic concept.

To deal with the problem mentioned above, we propose a novel active learning framework based on unsupervised learning. The basic idea is to “decouple” the relations between features and labels. We do not depend on instances’ labels in training as in supervised learning, but just form several high-quality clusters and let the machine “memorize” their labels. If a new sample cannot be well clustered to any of those formed clusters, it is considered as highly informative and thus selected out for human labeling.

In unsupervised learning, graph-theoretic clustering has recently attracted broad interest due to its intuitiveness and strong theoretical fundamentals [8], [12]. Among a great many graph-theoretic clustering algorithms, spectral clustering and dominant-set clustering are the most promising two. They complement each other in several aspects so we combine them in a hierarchy fashion to construct a hierarchical graph-theoretic clustering active learning system (HIGCALs).

The paper is organized as follow. In Section II, we introduce the two graph-theoretical clustering algorithms. In Section III, details of the proposed active learning framework and HIGCALs are expatiated. Experiment results are shown in Section IV. In Section V, we conclude our work on active learning.

## II. GRAPH-THEORETIC CLUSTERING

### A. Notations

In general, a set of data points  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  can be recognized as an undirected edge-weighted graph  $G = (V, E)$ , where  $V = \{v_1, \dots, v_n\}$  is the vertex set and  $E \subseteq V \times V$  is the weighted edge set with edge-weight  $w_{ij}$  reflecting the *similarity* between sample  $i$  and sample  $j$ :  $w_{ij} = \exp(-d(\mathbf{x}_i, \mathbf{x}_j)/(2\sigma^2))$ .  $d(\cdot, \cdot)$  is a kind of distance measure and Euclidean distance is commonly used. By convention, a symmetric affinity matrix  $\mathbf{A} = (a_{ij})$  is used to represent a graph  $G$ , where

$$a_{ij} = \begin{cases} w_{ij} & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

### B. Dominant-set Clustering

Dominant set is a novel combinatorial concept in graph theory proposed by M. Pavan *et al* [6]. It simultaneously emphasizes on *internal homogeneity* and *external inhomogeneity* thus can be considered as a great definition of “cluster”.

This work is partly supported by NSFC (Grant No. 60373046) and Natural Science Foundation of Beijing (Grant No. 4041004).

Wei Hu and Weiming Hu is with National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences {whu, wmmhu}@nlpr.ia.ac.cn

---

Algorithm 1: DSC

Input : Affinity matrix for  $k$ th iteration  $\mathbf{A}^k$

1. If  $\mathbf{A}^k$  is empty, return *NULL*
2. Calculate the local solution of (1) by (2):  $\mathbf{u}^k$  and  $f(\mathbf{u}^k)$
3. Get the dominant set:  $S^k = \Omega_{\mathbf{u}^k}$
4. Split out  $S^k$  from current graph and get a smaller graph with new affinity matrix  $\mathbf{A}^{k+1}$
5. Return:  $\text{DSC}(\mathbf{A}^{k+1}) \cup \{S^k, \mathbf{u}^k, f(\mathbf{u}^k)\}$

Output:  $\bigcup_{l=k}^K \{S^l, \mathbf{u}^l, f(\mathbf{u}^l)\}$

---

TABLE I  
DOMINANT-SET CLUSTERING ALGORITHM

---

Algorithm 2: DSFA

Input : Affinity vector  $\mathbf{a} \in \mathbb{R}^{n \times 1}$ ,  $\bigcup_{k=1}^K \{S^k, \mathbf{u}^k, f(\mathbf{u}^k)\}$

1.  $m^k = \frac{|S^k|-1}{|S^k|+1} \left( \frac{\mathbf{a}^T \mathbf{u}^k}{f(\mathbf{u}^k)} - 1 \right)$  for all  $k \in \{1, \dots, K\}$
2.  $k^* = \arg \max_k m^k$
3. If  $(m^{k^*} \leq 0)$   $k^* = 0$
4. Return:  $(k^*, m^{k^*})$

Output:  $k^*, m^{k^*}$

---

TABLE II  
DOMINANT-SET FAST ASSIGNMENT ALGORITHM

Consider the following quadratic program:

$$\begin{aligned} \max \quad & f(\mathbf{u}) = \mathbf{u}^T \mathbf{A} \mathbf{u} \\ \text{s.t.} \quad & \mathbf{u} \in \Delta \end{aligned} \quad (1)$$

where  $\mathbf{A}$  is the affinity matrix and  $\Delta = \{\mathbf{u} \in \mathbb{R}^n : \mathbf{u} \geq 0 \text{ and } \sum_{i=1}^n u_i = 1\}$ . Let  $\mathbf{u}^*$  denote a strict local solution of the above program;  $\Omega_{\mathbf{u}} = \{i : u_i > 0\}$  denote the support of  $\mathbf{u}$ . It has been proved in [6] that  $\Omega_{\mathbf{u}^*}$  is equivalent to one dominant set of the graph. So after a weighted graph is constructed, its dominant set can be searched out via solving program (1) and finding the support of its strict local solution. The local maximum  $f(\mathbf{u}^*)$  indicates the ‘‘cohesiveness’’ of the corresponding dominant-set cluster.

*Replicator equation* can be used to solve program (1):

$$u_i(t+1) = u_i(t) \frac{(\mathbf{A} \mathbf{u}(t))_i}{\mathbf{u}(t)^T \mathbf{A} \mathbf{u}(t)} \quad (2)$$

Several great dynamics properties (2) has been described in Theorem 2 in [6]. And its little computational demand is another great advantage compared to other graph-theoretic clustering algorithms where eigensolution is an indispensability.

Then there comes the dominant-set clustering algorithm shown in Table I. It is an iterative bipartition procedure, where a dominant set is split out from current graph in every iteration. Notice that the number of formed clusters  $K$  is automatically determined.

M. Pavan *et al* also made an out-of-sample extension for dominant-set clustering [7]. We just use the idea to examine a new sample  $\mathbf{x}^{\text{new}}$ . The algorithm is shown is Table II.  $\mathbf{a}$  is an affinity vector containing the similarities between  $\mathbf{x}^{\text{new}}$  and  $n$  existing samples.  $m^k$  is the membership of  $\mathbf{x}^{\text{new}}$  related to cluster  $S^k$ . If all the  $m^k$  are less than zero,  $\mathbf{x}^{\text{new}}$  is recognized

---

Algorithm 3: MSC

Input : Embedding matrix  $\mathbf{E} \in \mathbb{R}^{n \times K}$

% Step 1.2 initialize a orthogonal matrix  $\mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_K]$

1. Randomly Select  $i \in \{1, \dots, N\}$ , set  $\mathbf{r}_1 = [\mathbf{E}_{i1}, \dots, \mathbf{E}_{iK}]^T$  and  $\mathbf{c} = \mathbf{0}_N$
2. For  $k = 2, \dots, K$ 
  - a)  $\mathbf{c} = \mathbf{c} + \text{abs}(\mathbf{E} \mathbf{r}_{k-1})$
  - b)  $\mathbf{r}_k = [\mathbf{E}_{i1}, \dots, \mathbf{E}_{iK}]^T$ , where  $i = \arg \min_j \mathbf{c}(j)$
3. Set convergence indication  $\phi^{(0)} = 0$
4. For  $t = 1, \dots, T$ 

% Step a)~b) find the optimal discrete solution  $\mathbf{P}$ , step c)~f) find the optimal orthogonal matrix  $\mathbf{R}$ :

  - a)  $\mathbf{P} = \mathbf{E} \mathbf{R}$ ,  $\mathbf{X} = \mathbf{0}^{n \times K}$
  - b) For  $i = 1, \dots, N$ 
    - i)  $l = \arg \max_{k \in \{1, \dots, K\}} \mathbf{P}_{ik}$
    - ii)  $\mathbf{X}_{il} = 1$
  - c) SVD decomposition:  $\mathbf{X}^T \mathbf{P} = \mathbf{U} \mathbf{\Omega} \mathbf{V}$
  - d)  $\phi^{(t)} = \text{trace}(\mathbf{\Omega})$
  - e) If  $|\phi^{(t)} - \phi^{(t-1)}| < \theta_\phi$ , stop
  - f)  $\mathbf{R} = \mathbf{V} \mathbf{U}^T$
5. Return:  $\mathbf{P}$

Output: Partition Indication Matrix  $\mathbf{P} \in \mathbb{R}^{n \times K}$

---

TABLE III  
MULTICLASS SPECTRAL CLUSTERING ALGORITHM

---

Algorithm 4: SCEE

Input : Affinity vector  $\mathbf{a} \in \mathbb{R}^{n \times 1}$ , Affinity matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , Diagonal eigenvalue matrix  $\mathbf{\Lambda} \in \mathbb{R}^{K \times K}$ ,  $\mathbf{Z} \in \mathbb{R}^{n \times K}$ ,  $\mathbf{E} \in \mathbb{R}^{n \times K}$

1. For all  $i = \{1, \dots, n\}$ , calculate
 
$$\mathbf{w}(i) = \mathbf{a}(i) \left( \sum_{k=1}^n \mathbf{A}_{ik} + \mathbf{a}(i) \right)^{-1/2}$$
2.  $r = \frac{n+1}{n} \left( \sum_{k=1}^n \mathbf{a}(k) \right)^{-1/2}$ 

% Calculate the embedding  $\mathbf{y}$  of the new sample  $\mathbf{x}^{\text{new}}$ :

  - a)  $\mathbf{y} = r \mathbf{w}^T \mathbf{Z} \mathbf{\Lambda}^{-1} \in \mathbb{R}^{1 \times K}$
  - b) Normalization:  $\mathbf{y}(i) \leftarrow \mathbf{y}(i) / \left( \sum_j \mathbf{y}^2(j) \right)^{1/2}$
3. Return  $\mathbf{E}^{\text{new}} = [\mathbf{E}; \mathbf{y}]$

Output: New embedding matrix  $\mathbf{E}^{\text{new}} \in \mathbb{R}^{(n+1) \times K}$

---

TABLE IV  
SPECTRAL CLUSTERING EMBEDDING EXTENSION ALGORITHM

as an outlier. Otherwise,  $\mathbf{x}^{\text{new}}$  is considered belonging to the cluster with largest  $m^k$ .

### C. Spectral Clustering

Normalized cuts or spectral clustering is a stereotype of graph-theoretic clustering [8], [11], [5]. Most spectral clustering algorithms contain the following three steps:

1. Calculate the diagonal degree matrix  $\mathbf{D}$  for the affinity matrix  $A$  with  $\mathbf{D}_{ii} = \sum_{j=1}^n \mathbf{A}_{ij}$  and calculate matrix  $\mathbf{L} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ .
2. Calculate the  $K$  largest eigenvalues and eigenvectors  $\mathbf{z}_1, \dots, \mathbf{z}_K$  of  $\mathbf{L}$ , get the diagonal eigenvalue matrix  $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$  and matrix  $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_K] \in \mathbb{R}^{n \times K}$
3. Calculate the embedding matrix  $\mathbf{E}: \mathbf{E}_{ij} = \mathbf{Z}_{ij} / \left( \sum_j \mathbf{Z}_{ij}^2 \right)^{1/2}$ . Each row of  $\mathbf{E}$  can be considered as an embedding of the original sample.

Among multiclass spectral clustering algorithms, the one proposed in [11] is outstanding because of its efficiency and effectiveness. We show this algorithm in Table III. It

relaxes the discrete optimization and try to find the optimal rotation alignment and optimal discrete approximation alternatively. In practice, the algorithm converges after only several iterations. The output is an indication matrix. Each row corresponds to one sample and has only one position with value 1, indicating the cluster that sample belongs to.

Bengio *et al* in [1] made an excellent out-of-sample embedding extension for spectral clustering, which is shown in Table IV. When a new sample  $\mathbf{x}^{\text{new}}$  comes, its embedding  $\mathbf{y}$  and the new embedding matrix  $\mathbf{E}^{\text{new}}$  can be easily calculated. After that, we call algorithm *MSC* to finally determine which cluster  $\mathbf{x}^{\text{new}}$  belongs into.

### III. HIERARCHICAL GRAPH-THEORETIC CLUSTERING ACTIVE LEARNING SYSTEM

#### A. Overview of our active learning framework

To solve the semantic gap problem in active learning, we propose a novel active learning framework based on unsupervised learning. Our active learning framework takes on a hierarchical fashion. Different clustering algorithms complementary with each other are put in different layers to form many high-quality clusters in each layer. Then clusters are given labels based on their purity and cohesiveness and the labels are simply memorized. During clustering, we do not need label information or do not try to find a mapping  $f: \mathcal{X} \rightarrow \{1, \dots, L\}$  from feature space  $\mathcal{X}$  to a finite label set  $\{1, \dots, L\}$ . So in this meaning, features and labels are “decoupled” to avoid semantic gap problem.

When a new sample comes, we examine it at different layers sequentially. If it can be well clustered to a formed cluster, the procedure stops and it is endowed with the same label as that cluster. If the sample cannot be well clustered at any layer, it is considered to be highly informative and selected out to be labeled by humans.

#### B. Quality of a cluster

As mention above, “quality” of a cluster is an important concept in our active learning framework. If a cluster has low quality, there is no meaning to assign it a fixed label and let the machine memorize it. For dominant-set clusters, “cohesiveness”(see II-B) is a good index of quality. But for other kinds of clusters, we have to use other indices. Entropy seems to be a good choice, but in our work, we use a simpler one, which we call “purity”.

Assume that  $N$  labeled samples have been clustered into  $K$  clusters. In cluster  $S^i$ , there are  $n_{S^i}(l)$  samples having label  $l$ . Define an “importance factor” of label  $l$  w.r.t cluster  $S^i$  as follow:

$$if_{S^i}(l) = \frac{n_{S^i}(l)}{N_{S^i}} \times \frac{n_{S^i}(l)}{N} \quad (3)$$

where  $N_{S^i}$  is the size of cluster  $S^i$ . This definition borrows ideas from *TF\*IDF* in document analysis. It means that if label  $l$  has many supporters in cluster  $S^i$  and at the same time has a relatively large distribution on  $S^i$  compared to other  $K - 1$  clusters,  $l$  is of much importance to  $S^i$ .

Let  $if_{S^i}(l_1)$  and  $if_{S^i}(l_2)$  denote the two largest components of vector  $if_{S^i}$ . Define “purity” of  $S^i$  as:

$$p_{S^i} = \frac{if_{S^i}(l_1)}{if_{S^i}(l_2)} \quad (4)$$

Large  $p_{S^i}$  means that label  $l_1$  predominants over  $S^i$  so  $S^i$  can be considered pure and be labeled  $l_1$ .

#### C. HIGCALs

We implement a two-layer graph-theoretic clustering active learning system based on the proposed framework. We choose dominant-set clustering and spectral clustering respectively for the first layer and the second layer, because they complement each other in the two following aspects:

- **Combinatorial Structure.** Dominant set can be viewed as a great definition of “cluster” because it emphasizes on *internal homogeneity* and *external inhomogeneity* simultaneously. While the combinatorial structure defined in spectral clustering is less intuitive but more complicated. Due to the difference, dominant-set clustering can obtain clusters of extremely high quality when current graph is large, but when the graph is relatively small, spectral clustering outperforms dominant-set clustering in finding high-quality clusters.
- **Computational Complexity.** Dominant-set clustering is an iterative bipartition procedure and most dominant sets are very small compared to the size of the whole graph, so generally dominant-set clustering consume more time than spectral clustering. But for out-of-sample extension, spectral clustering has to do SVD decomposition to finally determine which cluster a new sample falls into, which is more time consuming than dominant-set clustering.

HIGCALs works on two phases: *Initialization* phase and *Functioning* phase. Now we introduce these two phases in details.

1) *Initialization Phase:* In this phase, the architecture of HIGCALs is constructed, which is shown in Figure 1. First, the labeled samples are randomly split to  $D$  smaller sets. On each sample set  $d$ , we run algorithm *DSC* (see Table I) to get many high-quality dominant set clusters  $\{S_d^1, \dots, S_d^{K_d}\}$ . These clusters are labeled with  $l = \arg \max_l if_{S_d^k}(l)$ . The importance factors  $if_{S_d^k}$  can be calculated using (3). The set  $\{S_d^1, \dots, S_d^{K_d}\}$  is called a “dominant-set clustering machine” (DSCM). So after dominant-set clustering, we obtain  $D$  DSCMs in the first layer. Then the samples in bad-quality dominant set clusters are all collected to form a new sample set, which we call “spectral clustering machines” (SCM). We put the SCM in the second layer and calculate its  $\Lambda$ ,  $\mathbf{Z}$  and  $\mathbf{E}$  as mention in section II-C. The algorithm for initialization phase is shown in Table V, where superscripts represent the number of the dominant-set cluster and subscripts the number of the DSCM. Notice we use both “purity”  $p_{S_d^k}$  and “cohesiveness”  $f(\mathbf{u}_d^k)$  to determine the quality of cluster  $S_d^k$ .

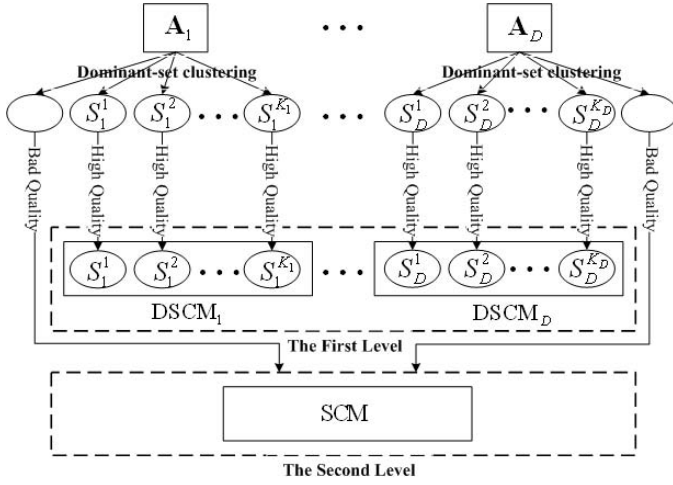


Fig. 1. Illustration of the Initialization phase of HIGCALs

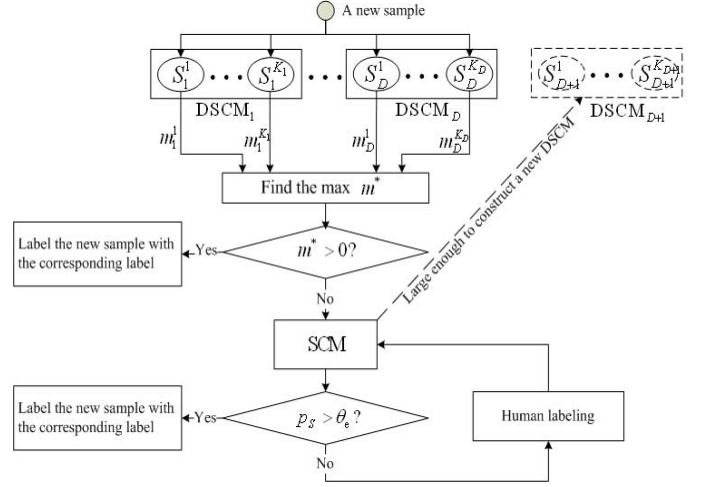


Fig. 2. Illustration of the Functioning phase of HIGCALs

**Algorithm 5: Initialization**

Input : Total number of labeled data  $N$ , Initial affinity matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$ , Size of each DSCM  $N_D$

1.  $D = N/N_D$ ;  $\text{SCM} = \emptyset$ ;  $\text{DSCM}_d = \emptyset, \forall d$
2. Randomly split  $\mathbf{A}$  to get  $\mathbf{A}_1, \dots, \mathbf{A}_D \in \mathbb{R}^{N_D \times N_D}$
3. For  $d = 1, \dots, D$ 
  - a)  $\text{DSC}(\mathbf{A}_d)$
  - b) For  $k = 1, \dots, K_d$ 
    - i) Calculate  $p_{S_d^k}$  using (3) and (4)
    - ii) If  $(p_{S_d^k} < \theta_p)$  or  $(f(\mathbf{u}_d^k) < \theta_f)$   
 $\text{SCM} = \text{SCM} \cup S_d^k$
    - iii) Else  
Label  $S_d^k$  and  $\text{DSCM}_d = \text{DSCM}_d \cup S_d^k$

4. Calculate  $\mathbf{A}^{\text{SCM}}, \mathbf{Z}^{\text{SCM}}$  and  $\mathbf{E}^{\text{SCM}}$   
Output:  $\{\text{DSCM}_1, \dots, \text{DSCM}_D\}, \text{SCM}$

TABLE V  
ALGORITHM FOR INITIALIZATION PHASE

2) *Functioning Phase*: The task of the functioning phase is to test new samples and update the system automatically. Figure 2 illustrate the procedure. When a new sample  $\mathbf{x}^{\text{new}}$  comes to the system, it is firstly fed to all DSCMs in the first layer. Algorithm *DSFA* (see Table II) is used in each DSCM to determine the largest membership. If all the memberships output by DSCMs are less than 0, which means  $\mathbf{x}^{\text{new}}$  is treated as an outlier by the first layer, it is fed to the second layer. Otherwise, it is assigned to the cluster with the largest membership among all DSCMs and owns the same label as that of the cluster. In the second layer, we first calculate  $\mathbf{x}^{\text{new}}$ 's embedding using algorithm *SCEE* (see Table IV), and cluster  $\mathbf{x}^{\text{new}}$  in SCM via algorithm *MSC* (see Table III). Then the purity of the cluster which  $\mathbf{x}^{\text{new}}$  falls into is examined. If it is smaller than a given threshold, which means it is still not secure to label  $\mathbf{x}^{\text{new}}$  in the second layer, we consider that  $\mathbf{x}^{\text{new}}$  is highly informative and should be selected out for humans to label. After manual labeling,  $\mathbf{x}^{\text{new}}$  with its label is fed back to SCM to improve the discriminating capability of SCM. When SCM is larger than a certain threshold, algorithm

**Algorithm 6: Functioning**

Input : A new sample  $\mathbf{x}^{\text{new}}, \{\text{DSCM}_1, \dots, \text{DSCM}_D\}, \text{SCM}$

1. Calculate the affinity vector of  $\mathbf{x}^{\text{new}}$  to all DSCMs:  $\mathbf{a}_d, \forall d$
2. For  $d = 1, \dots, D$ 

$$(k_d, m^{k_d}) = \text{DSFA}(\mathbf{a}_d, \bigcup_{k=1}^{K_d} \{S_d^k, \mathbf{u}_d^k, f(\mathbf{u}_d^k)\})$$
3.  $d^* = \arg \max_d m^{k_d}$
4. If  $(m^{k_{d^*}} > 0)$   
Assign  $\mathbf{x}^{\text{new}}$  with label  $k_{d^*}$
5. Else
  - a) Calculate the affinity vector of  $\mathbf{x}^{\text{new}}$  to SCM:  $\mathbf{a}$
  - b)  $\mathbf{E}^{\text{new}} = \text{SCEE}(\mathbf{a}, \mathbf{A}^{\text{SCM}}, \mathbf{\Lambda}^{\text{SCM}}, \mathbf{Z}^{\text{SCM}}, \mathbf{E}^{\text{SCM}})$
  - c)  $\mathbf{P} = \text{MSC}(\mathbf{E}^{\text{new}})$
  - d) From  $\mathbf{P}$ , determine the cluster  $\mathbf{x}^{\text{new}}$  falls into, and calculate its purity  $p$  using (4)
  - e) If  $(p > \theta'_p)$   
Assign  $\mathbf{x}^{\text{new}}$  with the corresponding label
  - f) Else
    - i) Present  $\mathbf{x}^{\text{new}}$  to human labeler
    - ii) Add the labeled  $\mathbf{x}^{\text{new}}$  to SCM, recalculate  $\mathbf{A}^{\text{SCM}}, \mathbf{\Lambda}^{\text{SCM}}, \mathbf{Z}^{\text{SCM}}, \mathbf{E}^{\text{SCM}}$
    - iii) If  $(s == \text{size}(\text{SCM})) > \theta_s$ ,  
Initilization( $\mathbf{A}^{\text{SCM}}, s, s$ )

TABLE VI  
ALGORITHM FOR FUNCTIONING PHASE

*Initialization* (see Table V) is called to construct an additional DSCM for the first layer and a new SCM for the second layer. The algorithm for functioning phase is shown in Table VI.

*D. Additional comments on HIGCALs*

In HIGCALs, we exploit dominant-set clustering for the first layer and spectral clustering for the second layer. Why not in reverse? There are several reasons.

Firstly, as we mentioned earlier, dominant-set clustering can generate clusters of extremely high quality when the current graph is large, due to the outstanding definition of dominant set. So compared to SCM, DSCMs contain clusters of much higher quality. Then obviously, DSCMs should be put in the first layer to make HIGCALs most efficient in

	NORMAL	DOS	U2R	R2L	PROBE	Total
Number in training set	97278	391458	52	1126	4107	494021
Number in each subset	1000	1500	52	500	500	3552
Total number for Initialization	5000	7500	52	1126	2500	16178
Ratio	5.14%	1.92%	100%	100%	60.87%	3.27%

TABLE VII

NUMBERS OF DIFFERENT CLASSES OF SAMPLES IN THE TRAINING SET AND FOR INITIALIZATION PHASE

	NORMAL	DOS	U2R	R2L	PROBE	Total
Number in test set	60593	223298	39	5993	2377	292300
Number selected for human labeling	475	2019	6	573	23	3096
Ratio	0.78%	0.90%	15.38%	9.56%	0.97%	1.06%

TABLE VIII

NUMBERS OF DIFFERENT CLASSES OF SAMPLES IN THE TEST SET AND SELECTED FOR HUMAN LABELING

dealing with a new sample.

Secondly, dominant-set clustering is an iterative bipartition algorithm, so the number of clusters is automatically determined. In practice, the size of each dominant set cluster is relatively small and there are many dominant set clusters, several of which may share the same labels. From this angle, we can imagine that the feature space are mostly divided into lots of small but highly homogeneous pieces, and the labels are distributed among them. That coincides very well with the basic idea of the proposed active learning framework mentioned in III-A. For spectral clustering, the number of clusters should be fixed beforehand, and cannot be large due to computational consideration. Moreover, clusters generated by spectral clustering are not as good in quality as those by dominant-set clustering, except when the graph is not large enough. So spectral clustering can only be considered as a supplement of dominant-set clustering thus should be put in the second layer.

Thirdly, in Initialization phase, dominant-set clustering consumes more time than spectral clustering. But in Functioning phase, the former is much faster than the latter for testing a new sample. Beside, because dominant set clusters owns higher quality, the discriminative capability of DSCMs is much higher than that of SCM. So we put dominant-set clustering in the first layer to make sure the efficiency of our active learning system.

HIGCALs has several great properties. Firstly, the system architecture is very flexible. In our practical system, we just design two layers. But notice that there are no limits on the number of layers and we can easily add or remove layers, as long as different unsupervised learning algorithms to some extent complement each other in different layers. Moreover, we can use other clustering algorithms rather than graph-theoretical clustering only if they coincide well with the idea of unsupervision-based active learning. Secondly, HIGCALs can easily and automatically upgrade itself. Ability to upgrade is a very important aspect for evaluating an active learning system. Because during learning, valuable label information provided by human labelers will be frequently fed back to the system and only by upgrade the system can enhance its discriminative capability gradually. In those

active learning methods which exploits supervised learning strategy, the classifier(s) has to be entirely retrained from time to time. But in our system, previously formed DSCMs need not to be altered at all. To upgrade the first layer is only to add a new DSCM, which will cover a different subspace in the feature space. Upgrade by this ‘‘addition’’ fashion is more flexible and stable than that in existing active learning systems.

#### IV. EXPERIMENTS

Network intrusion detection is a right field where active learning can be applied. Firstly, network data are usually of huge size. It is hard to manually label a large amount of data for supervised learning. Secondly, network is always changing. We can not expect to label a large amount of samples once and for all. Active learning can help network managers find out novel attacks with high efficiency. Thirdly, the semantic gap is usually large in analyzing network activities. For these reasons, we use KDDCUP99, a real-world network intrusion detection data set to evaluate HIGCALs. There are five general types of samples in the data set, each of which contains several specific types. The numbers of samples of various types in the training set and the test set are respectively listed in Table VII and Table VIII.

##### A. Initialization phase

In the Initialization phase, we randomly select five subsets of samples to form five DSCMs from the training set. The numbers of different classes of samples in the training set and for Initialization phase are shown in Table VII. Notice that U2R and R2L samples have to be repeatedly used because there are so few samples of these two types in the training set. We have in total  $N = 16178$  pre-labeled samples to initialize HIGCALs, which is relatively few compared to the whole training set (only 3.3%). Algorithm *Initialization* (see Table V) is called to construct five DSCMs and one SCM. We set  $\theta_p = 100, \theta_f = 0.1$  in advance, which is obtained from experiments.

##### B. Active learning capability

We use the test set to evaluate the active learning capability of HIGCALs. All the 292300 samples in the test set is fed

	Normal	DOS	U2R	R2L	PROBE	
Normal	57304	1201	329	832	452	95.32%
DOS	9812	210749	221	55	442	95.24%
U2R	4	1	23	5	0	69.70%
R2L	173	99	2	5141	5	94.85%
PROBE	14	21	5	2	2312	98.22%
	85.14%	99.38%	3.97%	85.19%	72.00%	

TABLE IX  
CONFUSION MATRIX OF THE TEST SET

in a random order to HIGCALs, and algorithm *Functioning* (see Table VI) is called, where we set  $\theta'_p = 10$ ,  $\theta_s = 2000$ . The numbers of different classes of samples in the test set and selected out for human labeling are shown in Table VIII. We can see in the whole procedure, 3096 samples are selected out in all, which means that our system can reduce human efforts as much as  $494021/(16178 + 3096) \approx 26$  times. Notice that U2R and R2L get a much higher ratio. For U2R, samples in the training set and test set are both too few to affect the clustering in either layer of HIGCALs. For R2L, severe semantic gap is a fundamental reason although HIGCALs has to a large extent weakened it.

### C. Accuracy

The confusion matrix of the test set is shown in Table IX, where the numbers do not include the samples selected out for manual labeling. Two indices are commonly used to judge the accuracy of a network intrusion detection method. One is *detection rate*:

$$DR = \frac{\text{\#Detected Attacks}}{\text{\#All Attacks}}$$

And the other is *false positive rate*:

$$FPR = 1 - \frac{\text{\#Detected Normals}}{\text{\#All Normals}}$$

The DR and FPR of the proposed active learning system are 95.63% and 4.68% respectively. This result is competitive with some published ones mentioned in [3].

## V. CONCLUSION AND FUTURE WORK

In active learning, most researches are based on supervised learning strategy. But under the conditions that the labeled data are too few or the semantic gap is large, these existing active learning methods may not effectively reduce human intervention for data labeling. In this paper, a new active learning framework based on unsupervised learning has been proposed and an active learning system called HIGCALs implemented. This system has a two-layer structure, where two promising graph-theoretic clustering methods complement each other to boost the active learning capability. The architecture and upgrade of HIGCALs are both flexible and easy to implement. Experiments have shown that our framework can greatly reduce the workload of manual labeling while maintaining favorable accuracy.

There is still much work to do. Firstly, the thresholds such as  $\theta_p$ ,  $\theta_f$ ,  $\theta'_p$  and  $\theta_p$  are all pre-determined by small-scale

experiments. We are trying to design a mechanism by which the system can automatically learn and adaptively adjust these parameters or even discard some of them. Take  $\theta_s$  in Table VI for example. By this parameter, decision about when the SCM should be promoted to construct a novel DSCM is made in a “hard” way. But a softer strategy can be exploited. For instance, when the number of high-quality clusters formed in current SCM is larger than half of the average of those contained in one DSCM, the system should upgrade itself. Secondly, there is still not a standard way to evaluate and compare different active learning algorithms. The comprehensive comparison of HIGCALs and other active learning methods is on the future work. Thirdly, we will apply HIGCALs to some computer vision applications such as image retrieval and video annotation to further verify the potentials of HIGCALs.

## REFERENCES

- [1] Y. Bengio, J. Paiement, and P. Vincent. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Proceedings of Neural Information Processing Systems*, 2004.
- [2] S. C. H. Hoi and M. R. Lyu. A semi-supervised active learning framework for image retrieval. In *Processing of Computer Vision and Pattern Recognition*, volume 2, pages 20–25, 2005.
- [3] W. Hu and W. M. Hu. Network-based intrusion detection using adaboost algorithm. In *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence*, pages 712–717, 2005.
- [4] F. Jing, M. Li, H. J. Zhang, and B. Zhang. A unified framework for image retrieval using keyword and visual features. *IEEE Transactions on Image Processing*, 14(7):979–989, 2005.
- [5] P. P. L. Zelnik-Manor. Self-tuning spectral clustering. In *Proceedings of Neural Information Processing Systems*, 2005.
- [6] M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In *Proceedings of Computer Vision and Pattern Recognition*, volume 1, pages 18–20, 2003.
- [7] M. Pavan and M. Pelillo. Efficient out-of-sample extension of dominant-set clusters. In *Proceedings of Neural Information Processing Systems*, 2004.
- [8] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [9] G. Tur, R. E. Schapire, and D. Hakkani-Tur. Active learning for spoken language understanding. In *Processing of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 6–10, 2003.
- [10] L. Wang, K. L. Chan, and Y. P. Tan. Image retrieval with svm active learning embedding euclidean search. In *Processing of International Conference on Image Processing*, volume 1, pages 14–17, 2003.
- [11] S. X. Yu and J. Shi. Multiclass spectral clustering. In *Proceedings of Ninth IEEE International Conference on Computer Vision*, volume 1, pages 313–319, 2003.
- [12] D. Q. Zhang, C. Y. Lin, S. F. Chang, and J. R. Smith. Semantic video clustering across sources using bipartite spectral clustering. In *IEEE International Conference on Multimedia and Expo*, volume 1, pages 117–120, 2004.