Class Augmented Active Learning

Hong Cao Chunyu Bao Xiao-Li Li* Yew-Kwong Woon†

Abstract

Traditional active learning encounters a cold start issue when very few labelled examples are present for learning a decent initial classifier. Its poor quality subsequently affects selection of the next query and stability of the iterative learning process, resulting in more annotation effort from a domain expert. To address this issue, this paper presents a novel class augmentation technique, which enhances each class's representation which initially consists of only limited set of labelled examples. Our augmentation employs a connectivity-based influence computation algorithm with an incorporated decaying mechanism for the unlabelled samples. Besides augmentation, our method also introduces structure preserving oversampling to correct class imbalance. Extensive experiments on ten publicly available data sets demonstrate the effectiveness of our proposed method over existing state-of-the-art methods. Moreover, our proposed modules perform at the fundamental data level without any requirement to modify the well-established standard machine learning tools.

1 Introduction

Today, data in various forms proliferate in an extremely fast rate and this poses severe challenges on traditional data mining and machine learning algorithms for predictive modelling. Data labelling, an essential yet often the most laborious process in the training data preparation, now becomes too expensive and almost infeasible in many real-world scenarios. This is because labelling sufficient training data for learning algorithms requires a large amount of time and annotation effort from a domain expert. It is thus desirable to significantly reduce the human effort needed by intelligently selecting only a small subset of examples for an expert to label in a sequential manner through active learning. As illustrated in Fig. 1, active learning (AL), as a branch of machine learning, is such a process of guiding the sampling process iteratively by querying certain types of instances from a large unlabelled pool for an expert to label based upon limited existing labelled examples whose class labels are known. One main challenge is how to select the next best sample or a batch of samples in a learning cycle, which can best complement the current labelled data. As reviewed in [1], existing pool-based AL algorithms design and optimize their query selection strategies in different ways, such as the largest uncertainty (or informativeness) with respect to the current classification model [2] [3], the largest reduction in version space [4], the largest influence on the current model, the largest reduction of the expected error, the largest reduction of the variances of the predictive outcomes, as well as accounting both representativeness and informativeness through pre-clustering [5] or through designing a combined new criterion [6]. Here, pool-based AL refers to the case where only a limited number of samples are labelled while a large static pool of unlabelled samples are present, which is contrary to the stream-based AL scenario.

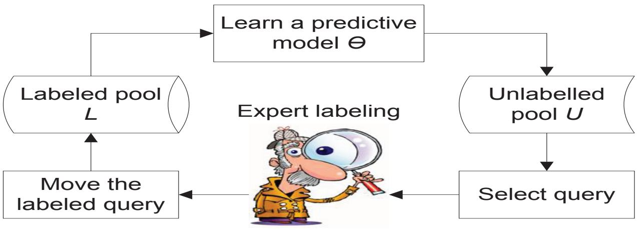


Figure 1: Active learning cycle

Though many AL algorithms have been developed, it remains challenging to apply AL algorithms in practical settings. The work in [7] discusses six types of challenges in AL, such as querying in batch mode, noisy expert labels, variable labelling cost, alternative query types, AL for multi-task learning and unknown model class. Besides such issues, the learning data can also be inherently class-imbalanced in their underlying distribution and so will be the uncovered labelled set. As the imbalance can undesirably bias in favor of the majority classes of large sizes, some AL works address this imbalance issue through selecting the samples only within the margin of the SVM boundary [8] or through guided learning [9], which controls the selection ratio of

*Data Analytics Department, Institute for Infocomm Research ($\mathrm{I^2R}$), A*STAR; Email: {hcao, cbao, xlli}@i2r.a-star.edu.sg; Address: 1 Fusionopolis Way, #21-01 Connexis (South Tower), Singapore 138632.

†EADS Innovation Works South Asia. Email: David.Woon @eads.net

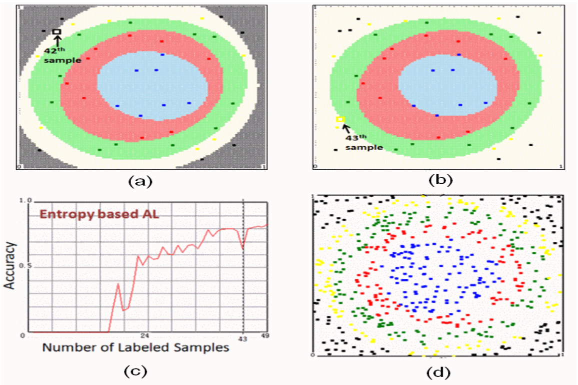


Figure 2: Labelling a new selected query (the 43th example) in (b) causes drastic change in the learnt class boundary regions (denoted by different colors) as compared with (a) as well as the decreased accuracy (from 80% to 66%), where the dots denote labelled samples. (c) Learning curve. (d) Distribution of the test ring dataset with each class on a ring with different diameters. Entropy based active learning [3] is used.

majority and minority class samples. Other practical issues also include active learning involving knowledge transfer cross multiple domains [10] and the cold start issue, where only very few labelled samples are initially present to kick off the AL process. To address the cold start issue, a recent work in [11] proposed an optimization algorithm to select the most representative data samples.

Knowing it is hard to meet all challenges simultaneously, in the paper, we limit our focus to AL in a two-class scenario, where only very limited set of labelled positive and negative samples are present initially in contrast to the large unlabelled pool. Our objective is to design an AL algorithm, that rises fast and steadily at the initial iterations to a desired accuracy level, which could be sufficient for practical usage. The fast rising rate at the initial phase is important as it minimizes annotation effort from the expert. The steadiness on the improvement of the learning curve also helps provide good feedback and foster the trust of the domain expert on the active learning process. In such a setting, we find that it is vital for our algorithm to meet two key challenges, namely cold start and class imbalance with insufficient minority-class representation. In particular, we highlight cold start as a significant issue, because it is ill-posed to train a good and stable probabilistic classification model using very limited set of labelled samples. When this model is subsequently used to select the next query in the unlabelled pool, its poor quality and instability could adversely affect the selection process, resulting in more human annotation effort. Fig. 2 shows one such example, where the multi-class classification performance decreases significantly with one additional training example, indicating that simply using the limited training examples only is hard to build a robust classifier. Such a scenario is common to binary classification as well.

We propose in this paper a new class-augmented active learning (CAAL) algorithm to address the above challenges. Our contribution is summarized below:

1. For the cold start issue, we design an augmentation algorithm to enhance each class's labelled sample set by searching for its likely class representatives from the large unlabelled sample set. To the best of our knowledge, this is the first class augmentation algorithm designed for active learning.

2. We introduce a structure preserving oversampling (SPO) algorithm to correct class imbalance and improve stability of the learning curve.

3. Through extensive experiments, we demonstrate effectiveness of our proposed method over standard active learning algorithms and some semi-active learning algorithms across 10 diverse publicly available datasets.

Note both our class augmentation and SPO modules operate at the fundamental data level and can be used in conjunction with various well-established standard learning tools. There is no requirement to modify such standard tools, which are often in the form of black boxes for practitioners. This is unlike the conventional active learning algorithms, e.g. [8], which requires customization of Support Vector Machines (SVM) algorithm, and the semi-active learning algorithms.

2 Our Proposed Method

Referencing to Fig. 1, we insert two new blocks, "class augmentation" and "imbalance correction", before the "learning of a predictive model" in our class-augmented active learning (CAAL) framework. In the following section, we detail these techniques.

2.1 Class Augmentation Given an input dataset $D = \{x\}$ and an instance $x \in D$, its r-neighbourhood density estimator is defined as [12]:

$$f_r(x) = \frac{N_r(x)}{N \cdot r} \tag{2.1}$$

where r is the radius, $N = |D|$ is the number of examples in D and $N_r(x) = |\{q \mid \|x - q\| \leq r, q \in D\}|$. Here $\|x - q\|$ is the Euclidean distance between x and q. For each instance x, we can use function $f_r(x)$ to estimate its density within a multi-dimensional sphere centered at x with a radius of r.

Let $D = L \cup U$, where L and U denote the current labelled and unlabelled sets, respectively. We discuss how to use the density estimator to augment labelled examples with the selected unlabelled examples or the virtual labelled examples.

For a labelled instance $x_0 \in L$, if $N_r(x_0) \geq J_r$, we consider x_0 is located in a reasonably dense space and its neighbours have direct connection to x_0. Here, J_r is user-defined with a typical value $J_r = 5$ [12].

Because the N_r neighbours are *directly connected* to x_0, we define a neighbourhood connection distance (or c-distance) of those neighbours as 1 to x_0 and denote these neighbours as $x_{1,1}$, $x_{1,2}$, ..., x_{1,n_1}. The list comprised of $\{x_{1,1}, x_{1,2}, ..., x_{1,n_1}\}$ is called a *connection list* of x_0. On the other hand, if $N_r(x_0) < J_r$, we consider x_0 is located in a low-density space and its neighbours have no connection to x_0.

Similarly, for each newly added neighbour $x_{1,i}$, we compute its $N_r(x_{1,i})$ (i=1, 2, ..., n_1). If $x_{1,i}$ is in high-density space, we add its neighbours which are not previously included to the second-level connection list, i.e. $x_{2,1}$, $x_{2,2}$, ..., x_{2,n_2}. Correspondingly, the c-distance between each of these newly added neighbours to x_0 is 2. The c-distance here is the number of connections to reach a new instance from a source instance through a density based connection path.

We iterate the expansion steps above to include those instances with c-distance = 3, 4, 5, ... to the connection list until no further expansion is possible. In this way, we construct a multi-distance connection list consisting of all the density-reachable samples for each labelled sample x_0 and this list can consist of both unlabelled instances and labelled instances. All the unlabelled instances in a connection list are considered as candidates to be selected for the class augmentation as they are directly density-reachable. If an unlabeled instance has no connection to the labelled instance x_0, its c-distance to x_0 will be ∞.

Note augmenting the limited labelled examples (x_0) with their densely connected unlabelled examples in a connectivity-based manner is important in our view because the data from one individual class (labelled or unlabelled) could be heterogeneously distributed in the space. However, samples of the same class are always likely clustered together with more apparent inter-connectivity than the samples from different classes. The connectivity-based augmentation allows our method to automatically take into account the local structures in the class data distribution.

We now explain the criterion to select the augmentation examples for each class.

Let x_i be a labelled instance with its label $l_i \in \{-1, 1\}$ and x_j be an unlabelled instance in the *connection list* of x_i. We let each labelled instance propagate its influence to the unlabelled instances in its connection list. We quantify the influence to x_j from the labelled instance x_i using a proposed weight computation formula below:

$$w_{j,i} = \frac{\beta^{sign(l_i)}(1-\alpha)^K}{dist(x_j, x_i)} \tag{2.2}$$

where $dist(x_j, x_i)$ denote the c-distance from x_j to x_i, $\alpha = N_l/N$, $\beta^+ = N_{l-}/N_{l+}$, $\beta^- = 1$ and $N_l = |L|$. N_{l-} and N_{l+} are the total number of labelled negative-class instances and positive-class instances, respectively. β^+ and β^- are normalization factors so that the influences from the two classes are balanced even in the case that the currently labelled examples from one class significantly outnumber those from the other class. $(1-\alpha)^K$ is a multiplier term controlling the amount of decay with reference to the current percentage of labelled samples α and a constant K controlling the rate of decay. The rationale is that our augmentation is important when the percentage of labelled instances is small, e.g. during the cold start; but it becomes less and less important when a growing percentage of unlabelled samples get labelled. A small K, e.g. $K = 0$, will result in a flat decay curve so that the augmentation strength controlled by $w_{j,i}$ remains strong even if a significant percentage of originally unlabelled data become labelled. On the other hand, a very large K can result in a steep decay curve such that insufficient augmented examples may be found even during the cold start. We choose $K = 3$ in our experiment after conducting sensitivity test.

Our influence computation formula in Eqn (2.2) is analogous to computing the electric potential induced by point charges in physics [13]. The labelled instances of positive and negative classes are equivalent to the positive and negative point charges. Our influence computed is equivalent to the electric potential induced collectively by the point charges in the multi-dimensional space. The main difference is that we adopt connection distance, i.e. c-distance, instead of Euclidean distance as it can take into account of the local class distribution. The connection distance is believed to work well in the common scenario that instances of the same class are clustered together and are easy to connect with each other; but instances of the opposite classes are difficult to connect due to the low-density class boundary. Such an assumption is similar to the cluster assumption [14], which has been well-received in the machine learning community as the foundation of many state-of-the-art graph-based learning algorithms.

Note that all the labelled instances which connect to x_j can influence the unlabelled example x_j. The aggregate influence to an unlabeled instance x_j from all the labelled examples is thus computed as follows:

(2.3)
$$w_j = \sum_i w_{j,i} = \left[\sum_{i=1}^{N_{l+}} \frac{\beta^+}{dist(x_j, x_{p_i})} - \sum_{k=1}^{N_{l-}} \frac{\beta^-}{dist(x_j, x_{n_k})} \right] (1-\alpha)^K$$

where $1 \leq p_i \leq N_{l+}$ is the index of the *i-th* positive labelled instance and $1 \leq n_k \leq N_{l-}$ is the index of the *k-th* negative labelled instance. Note in Eqn (2.3), only those labelled examples connected to x_j have the influence while those unconnected examples will have zero influence due to its their infinite c-distance.

The example x_j will be regarded as an augmented example to Class $sign(w_j)$ if its weight w_j satisfies:

$$|w_j| > w_0 \tag{2.4}$$

where w_0 is the median weight level, which is adaptively chosen using Algorithm 1. Before we present Algorithm 1, let us first define G_j as follows:

$$G_j = \sum_{i=1}^{N_{l+}} \frac{\beta^+}{dist(x_j, x_{p_i})} - \sum_{k=1}^{N_{l-}} \frac{\beta^-}{dist(x_j, x_{n_k})} \tag{2.5}$$

G_j is used in Algorithm 1 below to determine a context-dependent median weight level w_0.

Algorithm 1: Determining the median weight level w_0

Input: Labelled instances L, number of all instances N

1. Compute G_j for all labelled instances ($j = 1, 2, ..., N_l$) using Eqn (2.5). if $G_j \neq 0$, we put $|G_j|$ into a G-list. This excludes all unlabelled instances that do not connect to any of the labelled instances. Also, it is noteworthy to point out that Eqn (2.5) is similar to Eqn (2.3) except that the decay term is taken out.
2. Sort the elements in G-list from small to large.
3. Find the element G_{mid} in the middle of G-list and set $w_0 = G_{mid}$.

2.2 Class Imbalance Correction with Structure Preserving Oversampling (SPO) We propose to jointly correct class imbalance and to enhance distribution of the under-represented class using an SPO oversampling technique [15]. This imbalance correction is important to ensure stability of the active learning process as the distribution of the augmented learning dataset can be highly imbalanced and vary over the AL iterations. We recognize the oversampling technique in [15] well suits the needs to solve the cold start problem as it demostrated good oversampling performance in the scenario that the minority class is under-represented with a limited number of examples. We explain our oversampling algorithm below in brevity:

Given labelled examples from the minority class $P = \{x_{11}, x_{12}, ..., x_{1|P|}\}$ and those from the majority class $Q = \{x_{01}, x_{02}, ..., x_{0|Q|}\}$, where $x_{ij} \in \mathbf{R}^{n \times 1}$ and n is the feature dimension and $|P| < |Q|$, SPO performs oversampling in the following three key steps:

1. *Modeling minority class distribution*: Compute the mean $\bar{x}_1$ and the covariance $\mathbf{W}_P$ of the minority class. Then perform the eigen decomposition below:

$$\mathbf{D} = \mathbf{V}^T \mathbf{W}_P \mathbf{V} \tag{2.6}$$

where $\mathbf{D}$ is a diagonal matrix of eigenvalues with $d_1 \geq ... \geq d_j \geq ... \geq d_n$ for $n \geq j \geq 1$ and $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_j, ..., \mathbf{v}_n]$ is the corresponding eigenvector matrix.

2. *Regularize the eigenspectrum*: Owning to the limited data, the dominant eigenvalues in the eigenspectrum $\{d_j\}$ is usually reliable while the numerous trailing small eigenvalues can be erroneous. We use two-fold cross-validation to find the location M ($M < n$) that separates the reliable eigen spectrum region ($\leq M$) and the unreliable spectrum ($> M$) [15]. For each eigenvalue, we perform its regularization using

$$\begin{aligned} \hat{d_j} &= \quad d_j, \qquad if\ \ j < M \\ \hat{d_j} &= \lambda/(j+\mu), \quad otherwise \end{aligned} \tag{2.7}$$

where $\lambda = \frac{d_1 d_M (M-1)}{d_1 - d_M}$ and $\mu = \frac{M d_M - d_1}{d_1 - d_M}$ are eigen spectrum model parameters learnt from the reliable eigen spectrum. We then use the regularized eigenvalues to form a new diagonal matrix $\hat{\mathbf{D}}$.

3. *Generate synthetic examples with cleaning*: We first generate two vectors following two multivariate Gaussian distributions $N(\mathbf{0}_M, \mathbf{I}_M)$ and $N(\mathbf{0}_{n-M}, \mathbf{I}_{n-M})$. These vectors are concatenated to form a new vector $\mathbf{z}$. A synthetic minority-class sample is generated from $\mathbf{z}$ using

$$\mathbf{b} = \hat{\mathbf{D}}^{1/2} \mathbf{V}^T \mathbf{z} + \bar{x}_1 \tag{2.8}$$

We further check if the new synthetic sample is located in the space of the majority class Q and a cleaning procedure in [15] is used to remove these outlier samples. We repeat synthetic sample generation until the two classes are completely balanced with equal number of samples.

Table 1: Data Sets

Acronym	Data Name	Positive Class	Instances		Dim.
			+	-	
Steel	Steel Plates Faults	7	673	1268	27
RWine	Red Wine Quality	3	681	918	11
CMC	Contraceptive Method Choice	1	629	844	9
GCredit	German Credit	2	300	700	24
ImSeg	Image Segmentation	1	330	1980	18
BCancer	Breast Cancer Wisconsin	1	212	357	30
Pima	Pima Indians Diabetes	1	268	500	8
Wave	Waveform Database Generator	3	1696	3304	21
Digits	Pen-Based Recognition of Handwritten Digits	2	1144	9848	21
SLeaf	Swedish Leaf	1	75	1050	128

3 Experimental Results

3.1 Experimental Setup We conduct experiments on ten public datasets with nine from the UCI machine learning repository [16] and one from the UCR time series repository [17] with a good coverage of diverse domains. As we focus on binary classification, we use two rules to convert each multi-class dataset into a two-class dataset in a one-versus-others manner, where one of the existing classes is selected as the positive class and the remaining form the negative class. The first rule is that the positive class is smaller in size than the negative class to reflect practical scenarios. On top of the first rule, the second rule is to select the largest class as our positive class to ensure sufficient data are available for our simulation and comparison of different AL algorithms. Details of these two-class datasets are tabulated in Table I. Out of the ten datasets, “SLeaf”, “Digits” and “ImSeg” have the largest imbalance ratios of 14, 8.6 and 6, respectively and the remaining seven datasets have a mild imbalance ratio ranging from 1.3 to 2.3. Here, imbalance ratio is simply the ratio of negative class’s size over that of the positive class. In the preprocessing, we normalize each feature to the range of $[0, 1]$ to avoid the cases that some features outweight other features simply due to their large numerical range.

For each two-class dataset, we perform five-fold cross-validation for five times and report the average results. We use LibSVM with radial basis function kernel as our classification model with a default set of parameters [18]. In each run, only five positive instances and five negative instances are randomly selected as initially labelled instances. For fair comparison, we keep the ten initially labelled instances unchanged in our comparative simulation using other AL methods. Then in each AL cycle, one instance is selected from unlabelled pool based on the query selection result and its actual label is revealed to simulate the labelling process of the expert. We use GMean and average F-measure of positive and negative classes as evaluation measures, which are common performance evaluation measures [19] [20] for imbalanced data classification.

Table 2: Performance Comparison of CAAL, AL with entropy based query selection and AL with random query selection

Data Set	Max Cycle	Max GMean	Method	70%	75%	80%	85%
Steel	500	67.5%	CAAL	5	5	13	39
			Entropy	201	247	312	NA
			Random	185	201	312	NA
RWine	500	73.4%	CAAL	12	20	51	79
			Entropy	103	103	113	138
			Random	88	110	111	119
CMC	500	63.4%	CAAL	5	5	9	36
			Entropy	155	168	201	221
			Random	134	139	185	276
GCredit	500	71.7%	CAAL	7	14	39	76
			Entropy	198	228	257	447
			Random	199	250	309	452
ImSeg	500	97.5%	CAAL	5	5	6	9
			Entropy	58	77	99	119
			Random	26	32	33	42
BCancer	250	96.1%	CAAL	5	5	5	5
			Entropy	23	26	26	40
			Random	20	20	21	22
Pima	350	74.1%	CAAL	5	6	7	9
			Entropy	41	42	58	58
			Random	59	77	84	87
Wave	1000	91.1%	CAAL	10	12	13	17
			Entropy	32	40	41	54
			Random	29	49	51	88
Digits	1000	98.6%	CAAL	5	5	5	5
			Entropy	51	52	57	58
			Random	19	22	67	75
SLeaf	500	90.0%	CAAL	5	6	7	11
			Entropy	255	472	NA	NA
			Random	NA	NA	NA	NA

3.2 Results and Discussion

3.2.1 Comparison with Entropy-based and Random Query Selection Table II shows the performance comparison of CAAL with two well-known methods: AL with entropy-based query selection [3] and AL with random query. For binary classification, the margin-based query becomes equivalent to entropy-based query selection [1] [2]. Note that we show the Max GMean results in the third column of Table 2 as the reference, which are achieved when all unlabelled

Table 3: Performance Comparison of CAAL, Entropy+SPO (Ent+SPO) and Random+SPO (Ran+SPO)

Data Set	Max Cycle	Max GMean	Method	70%	75%	80%	85%
Steel	500	67.5%	CAAL	5	5	13	39
			Ent+SPO	90	128	177	252
			Ran+SPO	61	74	89	133
RWine	500	73.4%	CAAL	12	20	51	79
			Ent+SPO	110	112	145	161
			Ran+SPO	88	92	97	107
CMC	500	63.4%	CAAL	5	5	9	36
			Ent+SPO	27	43	59	78
			Ran+SPO	21	42	67	84
GCredit	500	71.7%	CAAL	7	14	39	76
			Ent+SPO	57	74	90	125
			Ran+SPO	27	34	43	49
ImSeg	500	97.5%	CAAL	5	5	6	9
			Ent+SPO	44	54	67	79
			Ran+SPO	31	35	39	61
BCancer	250	96.1%	CAAL	5	5	5	5
			Ent+SPO	45	46	57	57
			Ran+SPO	18	18	19	22
Pima	350	74.1%	CAAL	5	6	7	9
			Ent+SPO	75	77	100	115
			Ran+SPO	43	48	53	57
Wave	1000	91.1%	CAAL	10	12	13	17
			Ent+SPO	56	56	57	77
			Ran+SPO	19	19	20	22
Digits	1000	98.6%	CAAL	5	5	5	5
			Ent+SPO	16	22	24	34
			Ran+SPO	8	8	8	9
SLeaf	500	90.0%	CAAL	5	6	7	11
			Ent+SPO	57	57	66	72
			Ran+SPO	56	57	59	67

data become labeled and SPO is further used for imbalance correction to prepare the training data set. We find such results typically represent the best learning outcomes in our experiments. Columns 5 to 8 show the number of iterations required for GMean reaches 70% to 85% of the corresponding Max GMean. The results show that CAAL consistently requires fewer iterations than the two traditional AL algorithms, signifying apparent reduction of annotation effort from the domain expert. We have also observed similar outcomes when F-measure is used for the evaluation, where the results are accessible online at http://goo.gl/tVyZyn. Note that besides the F-measure results, one can also find our additional sensitivity analysis on the parameters K and Jr at the same online link. The analysis shows that our performance is insensitive to these parameters and there exist good ranges to choose their values.

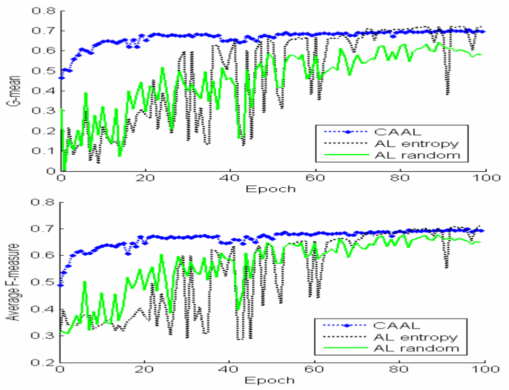


Figure 3: Performance comparison of CAAL, AL with entropy-based query selection and AL with random query selection on Pima

Fig. 3 shows the GMean and average F-measure based learning curves of the above-mentioned three methods. From the results, we observe good fluctuations visually of the GMean and F-measure learning curves using the benchmarking AL methods. On the other hand, the GMean and F-measure learning curves of CAAL are relatively stable with improved performance. In particular, the improvement of CAAL is apparent at the initial phase, which usually corresponds to the cold-start phase.

In Table 3, we further compare our CAAL with the scenario, where SPO is used for imbalance correction together with entropy-based or random query selection strategies. By comparing Table 2 and Table 3, we observe that SPO confidently improves the performance and stability of entropy-based and random query selection strategies in terms of reduced number of queries needed to reach the desired GMeans, i.e. 70%, 75%, 80% and 85% of the maximal achievable G-mean. This shows that adding SPO is also effective for improving the performance of entropy-based AL and the random query selection strategy. Even after adding SPO to the benchmarking AL methods, we still find in general, CAAL consistently requires much fewer queries to label than these methods for all the ten datasets. We attribute this to the novel class augmentation mechanism that we have proposed in CAAL.

3.2.2 Effectiveness Evaluation of Augmentation and SPO Procedures in CAAL We have also compared the learning curve of CAAL with the cases of CAAL without SPO and CAAL without class augmentation in Fig. 4. Clearly, we can see that by removing class augmentation, CAAL's performance decreases significantly in the initial phase. From 70 iterations onwards, we find that both curves saturate at about

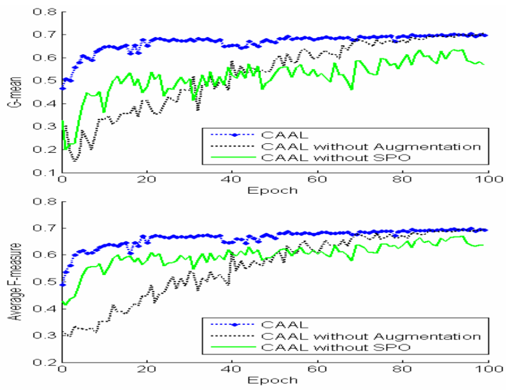


Figure 4: The performance of CAAL, CAAL without class augmentation and CAAL without SPO on Pima Indians Diabetes data.

the same level. Similar phenomenon can be observed from the results of other datasets showing the augmentation is effective to improve the performance at the initial phase. Fig. 5 also shows that the performance of CAAL without SPO is consistently lower than that of CAAL, suggesting the effectiveness of SPO in balancing the learning dataset with enhanced minority-class representation. Also, we observe that taking away either augmentation or SPO would introduce more severe observable fluctuations to the learning curves. The augmentation and SPO imbalance correction collectively improve the steadiness of CAAL's learning performance especially at the initial cold-start phase.

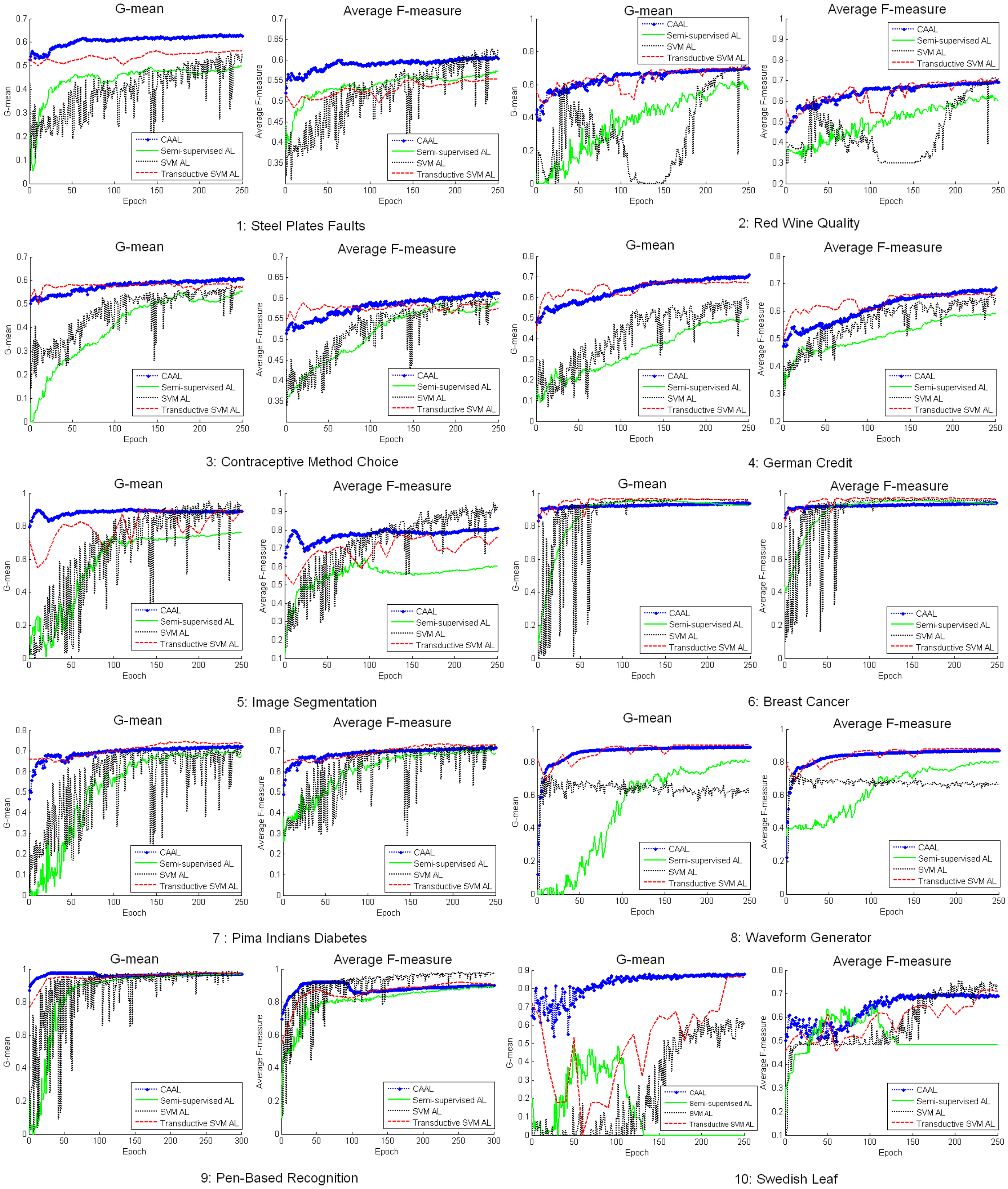
3.2.3 Comparison with Other Representatives of AL Algorithms For the completeness, we have also benchmarked CAAL with three well-known and relevant methods, which are SVM active learning (SVM-AL) by Tong et al. [4] [21]; semi-supervised AL algorithm proposed by Zhu et al. [22] and transductive-SVM based active learning with entropy-based query selection. It is worth to note that these methods are either customized to a particular standard learning tool or utilize the unlabelled data through semi-supervised learning or transductive experiment design. CAAL, on the other hand, operates at the data level without tieing to one standard learning tool.

Out of the three algorithms, SVM-AL selects the next query with an aim to best reduce the current version space. The later two are semi-supervised active learning algorithms, which utilize the unlabeled data for learning through semi-supervised learning algorithm or transductive learning algorithm. Fig. 5 compares the learning curves evaluated using the G-mean and the average F-measure for all the ten data sets. Fig. 6 are the average curves for the ten data sets. Clearly, we find that on average CAAL outperforms SVM-AL and Zhu's semi-active learning algorithm with good margins. The transductive-SVM achieved closer learning-curve performance to ours, but its running time is long for each active learning cycle. For example, for the CMC data, the computation time of one running (250 epochs) using CAAL is 19.94s, while that using transductive SVM (TSVM) takes 11206.3s to complete the same 250 epochs. This long processing time of TSVM is caused by the iterative and incremental procedure to incorporate the unlabelled examples into the modelling process and the hard optimization requirement, which is performed through integer programming. Compared with TSVM based AL, CAAL is faster with more stable performance improvement over the iterations. Though the processing time of CAAL are about two times of those required for SVM AL and semi-supervised AL to complete 250 epochs, CAAL's good and stable learning curve requires fewer iterations to reach the same level of performance. These features are highly desireable in a practical application setting.

4 Conclusions

We have presented in this paper a class-augmented active learning algorithm, which successfully addresses several practical challenges, namely cold start and class imbalance in the two-class active-learning setting. Through connectivity-based influence computation and using a decaying mechanism, our augmentation algorithm efficiently enriches the limited labelled dataset at the initial phase for building a decent classifier for the subsequent active learning process. When the percentage of labelled samples grows, our decaying mechanism also automatically reduces the augmentation strength. Our suggested imbalance correction technique, namely structure preserving oversampling, not only improves the learning performance in terms of GMean and F-measure, but also shows effectiveness of reducing the fluctuations of the learning curve, thus enhancing the learning stability. Experimentally, based on 10 public machine learning datasets across diverse domains, our proposed method consistently demonstrated significant performance gains to handle the cold start issue due to our class argumentation module. The experiments also show that our CAAL algorithm requires the labelling of only a small fraction of the number of samples required by other conventional active learning algorithms to achieve excellent performance at the initial phase.

As the various forms of data proliferate in an unprecedented rate, an active learning method that requires fewer labelled instances to kick off and requires significantly less annotation effort from domain expert-



1: Steel Plates Faults

2: Red Wine Quality

3: Contraceptive Method Choice

4: German Credit

5: Image Segmentation

6: Breast Cancer

7 : Pima Indians Diabetes

8: Waveform Generator

9: Pen-Based Recognition

10: Swedish Leaf

Figure 5: The G-mean and Average F-measure performance of CAAL, SVM AL, semi-supervised based AL and transductive SVM based AL on the ten data sets.

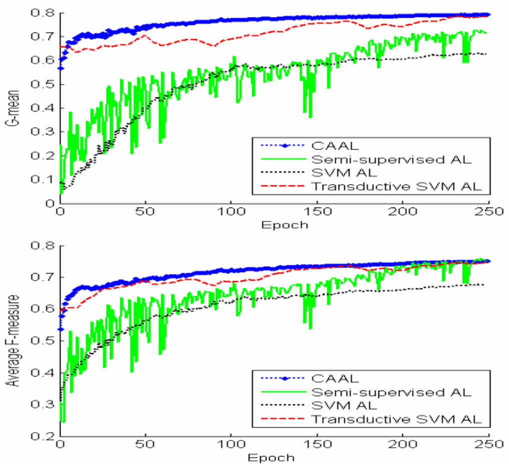


Figure 6: The average performance of CAAL, SVM AL, semi-supervised based AL and transductive SVM based AL on the ten data sets.

s become particularly important. Our simple solution proposed in this paper with remarkable results exhibits its desired property of addressing several practical active learning challenges. Our future research along this avenue will continue exploring class augmentation mechanisms that are adaptive to the inherent class data distribution. We also consider algorithm efficiency and scalability issues for performing active learning on complex datasets with ever-growing size, heterogeneity and attribute dimensionality.

References

[1] B. Settles, *Active Learning Literature Survey.* University of Wisconsin–Madison: Computer Sciences Technical Report 1648, 2010.

[2] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *Proc. of CAIDA*, 2001, pp. 309–318.

[3] D. Lewis and W. Gale, "A sequential algorithm for training text classifiers," in *Proc. of ICRDIR*, 1994, pp. 3–12.

[4] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," in *Proc. of ICML*, 2000, pp. 999–1006.

[5] H. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *Proc. of ICML*, 2004, pp. 623–630.

[6] S. Huang, R. Jin, and Z. Zhou, "Active learning by querying informative and representative examples," *Proc. of NIPS*, pp. 892–900, 2010.

[7] B. Settles, "From theories to queries: Active learning in practice," *Proc. of JMLR Workshop*, pp. 1–18, 2011.

[8] S. Ertekin, J. Huang, L. Bottou, and C. L. Giles, "Learning on the border: active learning in imbalanced data classification," in *Proc. of CIKM*, 2007, pp. 6–8.

[9] J. Attenberg and F. Provost, "Why label when you can search?: alternatives to active learning for applying human resources to build classificaton models under extreme imbalance," in *Proc. of ACM SIGKDD*, 2010, pp. 423–432.

[10] L. Li, X. Jin, S. Pan, and J.-T. Sun, "Multi-domain active learning for text classification," in *Proc. of ACM SIGKDD*, 2012, pp. 1086–1094.

[11] F. Nie, H. Wang, H. Huang, and C. H. Ding, "Early active learning via robust representation and structured sparsity," in *Proc. of IJCAI*, 2013, pp. 1572–1578.

[12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of KDD*, 1996, pp. 226–231.

[13] D. Griffiths, *Introduction to Electrodynamics (3rd. ed.).* Prentice Hall, 1998.

[14] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Proc. of NIPS*, vol. 16, p. 284, 2004.

[15] H. Cao, X.-L. Li, Y.-K. Woon, and S.-K. Ng, "SPO: Structure preserving oversampling for imbalanced time series classification," in *Proc. of ICDM*, 2011.

[16] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[17] E. Keogh, Q. Zhu, B. Hu, H. Y., X. Xi, L. Wei, and C. A. Ratanamahatana, "The UCR time series classification/clustering," 2011.

[18] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[19] H. Cao, X.-L. Li, Y.-K. Woon, and S.-K. Ng, "Integrated oversampling for imbalanced time series classification," *IEEE Trans. on Knowledge and Data Engineering*, 2013.

[20] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. on Knowledge and Data Engineering*, vol. 21(9), pp. 1263–1284, 2009.

[21] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, pp. 45–66, 2002.

[22] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. ICML Workshop*, 2003.