Knowl Inf Syst (2013) 35:249–283
DOI 10.1007/s10115-012-0507-8

REGULAR PAPER

A survey on instance selection for active learning

Yifan Fu · Xingquan Zhu · Bin Li

Received: 22 July 2011 / Revised: 2 March 2012 / Accepted: 17 March 2012 /
Published online: 6 June 2012

Abstract Active learning aims to train an accurate prediction model with minimum cost by labeling most informative instances. In this paper, we survey existing works on active learning from an instance-selection perspective and classify them into two categories with a progressive relationship: (1) active learning merely based on uncertainty of independent and identically distributed (IID) instances, and (2) active learning by further taking into account instance correlations. Using the above categorization, we summarize major approaches in the field, along with their technical strengths/weaknesses, followed by a simple runtime performance comparison, and discussion about emerging active learning applications and instance-selection challenges therein. This survey intends to provide a high-level summarization for active learning and motivates interested readers to consider instance-selection approaches for designing effective active learning solutions.

Keywords Active learning survey · Instance selection · Uncertainty sampling · Instance correlations

1 Introduction

Electronic data management systems have rapidly emerged in the past decades. All these systems computerize the data on operations, activities, and performance. For decision-making

In this paper, model and classifier are interchangeable terms.

Y. Fu (✉) · X. Zhu · B. Li
Centre for Quantum Computation and Intelligent Systems (QCIS),
Faculty of Engineering and Information Technology, University of Technology,
Sydney, NSW 2007, Australia
e-mail: fuyf939@gmail.com; Yifan.Fu@student.uts.edu.au

X. Zhu
e-mail: Xingquan.Zhu@uts.edu.au

B. Li
e-mail: Bin.Li-1@uts.edu.au

purposes, these systems typically rely on domain experts to manually analyze the database. Due to the rapid development of storage, sensing, networking, and communication technologies, recent years have witnessed a gigantic increase in the amount of daily collected data. As a result, it becomes rapidly difficult, or impossible, to manually extract useful knowledge from huge amount of data. The need for automated mining and discovering knowledge from large-scale data, commonly referred to as Knowledge discovery and data mining (KDD), is widely recognized. Common approaches in KDD are to either (1) generate patterns without supervision, such as clustering [85,94], or (2) use some previously labeled instances to assist the pattern discovery process, such as supervised learning [86]. For the latter, labeled instances can integrate domain knowledge and therefore help generate models mostly suitable for prediction. To collect labeled samples,[1] the labeling process may be subject to little or no cost, such as the "spam" flag users marking on unwanted email messages. But for many sophisticated supervised learning tasks, sample annotation requires costly expert efforts, which raises significant issues for some large-scale or domain-specific problems as follows:

- *Fraud Detection* [50]. A banking expert needs to manually inspect each credit transaction to properly label a transaction as either a fraud or a normal transaction. With manual inspection and labeling, it may take an expert several years to inspect all transaction records in a month and annotate a small amount of fraud transactions.
- *Webpage Classification* [51]. When query results are returned by a search engine based on a specific keyword, we need to identify whether a web page is relevant to the keyword or not. It has been shown that less than 0.0001 % of all web pages have topic labels. Therefore, annotating thousands of web pages can be tedious and redundant.
- *Protein Structure Prediction* [8]. Protein structure prediction is to find a protein's secondary, tertiary, and quaternary structures from the protein's amino acid sequence. However, less than 1.2 % of all proteins have known structures. For a specific protein, it takes months for a crystallographer to identify its structure in wet lab experiments.

In classical supervised learning, training instances must be paired with class labels as supervised knowledge. This constraint makes the above applications applicable to solve learning problems on a small fraction of labeled data. Because limited labeled instances can hardly provide sufficient information to learn models with good generalization capability, some new learning paradigms have been proposed in the last decade to reduce the labeling cost without significantly compromising the model performance. Two most successful types are: (1) Semi-supervised learning [87,88], which directly utilizes unlabeled instances by taking into account the geometry of data distributions, such as clusters and manifolds, to propagate label information to neighboring data; and (2) Active learning [89,90], which selectively labels instances by interactively selecting most informative instances based on certain instance-selection criteria. Since we only focus on active learning in this survey, interested readers can refer to [64] for works related to semi-supervised learning.

In this article, we aim to provide a comprehensive survey on active learning from an instance-selection perspective, where the goal of active learning is to achieve high prediction accuracy by using as few labeled instances as possible [45]. Suppose we are given a small labeled sample set as well as a relatively large unlabeled sample set, and we are allowed to interactively label a portion of the unlabeled instances during the learning process. Active learning essentially solves the following problem: *how to select most critical instances from the unlabeled sample set for labeling such that a model trained on them can achieve the maximum prediction accuracy, compared to simple solutions such as randomly labeling the same number of instances*

[1] In this paper, samples and instances are interchangeable terms.

Generally speaking, including most informative instances to the labeled set can help improve the model performance with least labeling costs or reduce the computational cost for the succeeding mining procedures [78]. In practice, the informativeness of a sample can be assessed by using the uncertainty of the instances based on models trained from the current labeled sample set. If a sample's uncertainty is high, it implies that current models do not have sufficient knowledge in classifying the sample, and, presumably, including this sample into the training set can thus help improve the underlying models. Following this heuristic, the key challenge for active learning is to design proper uncertainty metrics to evaluate the utility of an unlabeled sample [10]. A large number of methods have been proposed to quantify and assess sample uncertainty in various ways. From an instance-selection perspective, these methods can be classified into the following two categories, with a progressive relationship.

1. **Utility metrics merely based on uncertainty of IID instances**: Methods in this category treat samples as independent and identically distributed (IID) instances, where the selection criteria only depend on the uncertainty values computed with respect to each individual instance's own information. Accordingly, one possible problem is that this type of approach may select similar instances in the candidate set, which results in redundancy in the candidate set. Take the toy data in Fig. 1 as an example, if only considering the uncertainty of the instances for labeling, we are likely to select a candidate set only containing the most uncertain instances from an individual sample's perspective, whereas these instances may contain redundant knowledge and therefore do not form an ideal candidate set (as shown in Fig. 1b).
2. **Utility metrics further taking into account instance correlations**: To take the sample redundancy into consideration, uncertainty metrics based on instance correlation utilizes some similarity measures to discriminate differences between instances. By uncovering inherent relationships between instances, the utility of the instances calculated by this scheme integrates sample correlations, through which a selected candidate set may not always contain the "most uncertain" instances. Whereas, together, the selected instances form an optimal candidate set by balancing the uncertainty and diversity. As shown in Fig. 1c, by considering sample diversity, the six selected candidate instances help generate the decision boundary, which is much closer to the true boundary, compared to Fig. 1b where only uncertainty is considered. From this example, we can see that "tradeoff between uncertainty and diversity" is an essential problem to address in active learning. When considering too much "uncertainty", we may select redundant instances, whereas when considering too much "diversity", we may lose many uncertain instances that are critical for forming the boundary.

Several papers have surveyed active learning with focuses on different active learning scenarios [44] or on different application domains [39,53]. To the best of our knowledge, there is no survey focusing on instance correlations and summarizing active learning from instance-selection perspective. So our paper intends to provide an in-depth study on how existing active learning methods explore uncertainties and correlations to select instances for labeling. Our main objective is to (1) summarize and categorize instance-selection methods and provide a big picture for active learning, and (2) compare and analyze the strengths and deficiencies of existing approaches, through which interested readers can propose solutions to further advance the research in the field.

The remainder of the survey is organized as follows. In Sect. 2, we provide an overview of active learning, including preliminary concepts and a multi-dimensional categorization of the existing methods from an instance-selection perspective. Section 3 investigates instance-selection methods merely based on IID instance uncertainty. Section 4 further studies methods

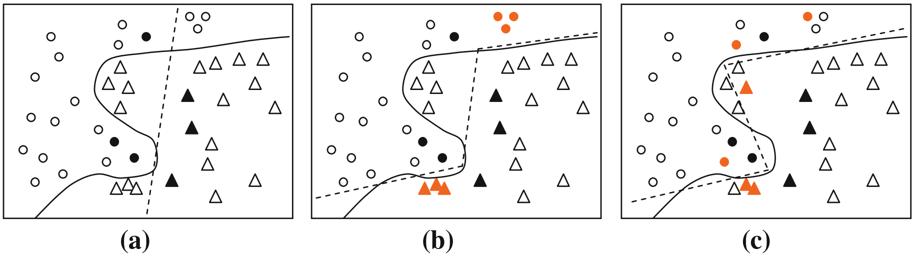


Fig. 1 A toy example to demonstrate the tradeoff between uncertainty and diversity for sample selection in active learning. *Circles* and *triangles* denote the instances from two classes, respectively; *solid circles* and *triangles* denote labeled instances and the rest denote unlabeled instances. The *solid lines* denote the true decision boundaries and the *dashed lines* denote the decision boundaries learned by the learners based on the selected instances. **a** Decision boundary learned from six labeled training instances. **b** By labeling six most uncertain instances, the learner refines its decision boundary, which becomes more approximate to the true decision boundary. **c** By taking sample diversity into consideration, a method chooses the most informative candidate instances with low redundancy between them, based on which the learned decision boundary is significantly improved, compared to the approach which considers uncertainty only

by taking into account instance correlations. In Sect. 5, we analyze the existing instance-selection methods by comparing their strength and weakness. Section 6 discusses emerging applications of active learning and instance-selection challenges therein. Finally, we conclude the survey in Sect. 7.

2 Active learning: preliminary and overview

2.1 Definitions and notations

Given a set of instances $\mathscr{D} = \{e^1, e^2, \ldots, e^n\}$, where each sample x^i is in a q- dimensional feature space $\mathscr{F}$ and an l dimensional label space $\mathscr{Y}$, *i.e.*, $e^i \in \mathscr{F} \times \mathscr{Y}$. Depending on the number of labels an instance contains, an instance can be divided into two types: a “single-label instance” and a “multi-label instance”.

Definition 1 Single-label Instance: For a single-label instance e^i, it can be denoted by $e^i = \{x^i, y^i\}$, where $x^i = \{f_1^i, f_2^i, \ldots, f_q^i\}$, and f_k^i denotes the kth feature value of e^i, and y^i denotes the class label of x^i.

Definition 2 Multi-Label Instance: For a multi-label instance, it can be denoted by $e^i = \{x^i, y_1^i, \ldots, y_l^i\}$, where $x^i = \{f_1^i, f_2^i, \ldots, f_q^i, \}$, and f_k^i denotes the kth feature value of e^i, and y_j^i denotes the jth class label of e^i.

In typical active learning scenarios, users are given an instance set D comprising a handful of labeled instance subset $D^L = \{(x^1, y^1), (x^2, y^2) \ldots, (x^n, y^n)\}$ and a relatively large amount of unlabeled instances $D^U = \{(x^1, ?), (x^2, ?) \ldots, (x^u, ?)\}$, with $D = D^L \bigcup D^U$. With limited labeling information, an accurate model can hardly be learned. To learn an accurate model, we need to label extra instances to get additional information. In an active learning environment, it is considered costly and time-consuming to label all instances in D^U. Alternatively, an active labeler utilizes evaluation metrics to measure instance utility and further selects instances with maximal utility values for labeling. There are two important concepts used in utility metrics: uncertainty metric and correlation. The former is an evaluation criterion to measure the uncertainty of each single instance, whereas the latter measures the "correlations" between instances.

Definition 3 Uncertainty Metric:. Given an unlabeled sample set D^U and a label space $\mathscr{Y}$, uncertainty metric is a function f_u mapping from the instance space, D^U or $D^U \times \mathscr{Y}$, to a real number space R where the "sample view" means the uncertainty metrics calculated based on the sample features, while "sample-label view" means the uncertainty metric calculated from both features and labels.

$$f_u : \begin{cases} D^U \mapsto R, & \text{sample view} \\ D^U \times \mathscr{Y} \mapsto R, & \text{sample-label view} \end{cases} \tag{1}$$

Most of the previous algorithms evaluate the uncertainty only from the sample view. More recently, research work has focussed on evaluating the uncertainty from both sample and sample-label views, which is considered to be more effective. In general, an uncertainty metric usually borrows information technology and statistical theory, such as "entropy" and "margin", to measure instance utility. Different functions used in the selection metrics prefer different types of instances. For example, an "entropy" function tends to select instances minimizing the log-loss of the model, whereas a "margin" function intends to choose the ones reducing the error rate by refining the decision boundary.

In addition to the above discussed "uncertainty" metric, one can also take "diversity" of the selection into consideration, which can be enabled by evaluating the correlation of the instances. Take Fig. 1 as an example, by uncovering the correlation between the instances, the selected labeling set in Fig. 1c helps generate a boundary much closer to the true decision boundary, compared to Fig. 1b where only uncertainty is considered. Accordingly, properly estimating correlation among instances is important for selecting most informative instances in active learning.

Definition 4 Correlation Metric: Given an unlabeled sample set D^U and a label space $\mathscr{Y}$, correlation metric is a function q_c used to measure the correlation between a pair of instances x_i and x_j, where the correlation between any instance pair, $q_c(x_i, x_j)$, can be defined from three views:

$$q_c : \begin{cases} D^U \times D^U \mapsto R, & \text{feature view} \\ \mathscr{Y} \times \mathscr{Y} \mapsto R, & \text{label view} \\ (D^U, \mathscr{Y}) \times (D^U, \mathscr{Y}) \mapsto R, & \text{both views} \end{cases} \tag{2}$$

By uncovering the pairwise correlation in the instance set, two instances with a large correlation value are considered similar to each other, while the two with a small value are different. With Eq. 2, we can define the correlation between x_i and any other instances in D^U, denoted by $q_c(x_i)$, which is the mean of the correlation $q_c(x_i, x_j)$ for all $j \neq i$

$$q_c(x_i) = \frac{1}{|D^U|} \sum_{x_j \in D^U / x_i} q_c(x_i, x_j) \tag{3}$$

Equation 3 represents the instance density in the unlabeled sample set. The larger the value $q(x_i)$, the higher the density around the instance x_i is. Therefore, the most representative instances in a set have the largest correlations. Intuitively, they are the centrex points with highest density. On the other hand, the instances with smallest correlation values are located at the edge of the set and are considered as outliers.

Based on the above "uncertainty metric" and "correlation metric" , the "utility metric" for active learning is defined as follows.

Definition 5 Utility Metric: Given an uncertainty metric f_u and/or a correlation metric q_c, utility metric is a function u used to evaluate the worth of labeling for unlabeled instances in D^U:

$$u = \begin{cases} f_u, & \text{if} \quad \text{not given } q_c \\ f_u \times q_c, & \text{if} \quad \text{given } q_c \end{cases} \tag{4}$$

The definition of f_u also explores the utility from two granularity levels: sample and sample-label pair. When utility metric integrates the correlation metric q, the instance utility is evaluated from both uncertainty and correlation views. As shown in Eq. 4, if f_u increases, uncertainty becomes larger, and so does u. However, only taking uncertainty into account results in a redundancy issue as introduced in Fig. 1, while we assess instance utility based on correlation that may select diverse instances. To this end, Eq. 4 is a trade-off function between the two views to assess instance utility.

Definition 6 Query Strategy: By choosing a certain utility metric, a query strategy evaluates the informativeness of unlabeled instances based on the prediction result of the current model (to calculate uncertainty) and/or data distributions (to calculate correlations), then selects the most informative instances for labeling.

With a specific query strategy, one can rank instances according to their utility values. The instances on the top of the queue are the most ambiguous ones for the current model, whereas the ones at the bottom of the queue are the most certain instances for the model. The top υ (where υ is the size of an optimal sampling subset) forms a maximal utility subset to be included in the training set. Following this approach, general procedures for active learning process are described in Algorithm 1.

Algorithm 1 General Process of Active Learning

```
Require: Initial labeled instance set $D^L$, Unlabeled instance set $D^U$, size of the training set m
Ensure: Model $\Theta$
while training size $\leq$ m do
  $\Theta \leftarrow$ learn a model based on $D^L$;
  $D^U \leftarrow D \setminus D^L$;
  for each instance $x_i$ in the $D^U$ do
    $u_i \leftarrow u(x_i, \Theta)$;
  end for
  $x^\star \leftarrow \arg\max_i (u_i)$;
  $D^L \leftarrow D^L \bigcup x^\star$;
  $D^U \leftarrow D^U \setminus x^\star$;
  $\Theta \leftarrow$ update the model based on $D^L$;
end while
```

In Algorithm 1, a model is trained from the initial small labeled training set D^L. After that, all instances in the unlabeled pool D^U are queried by the learner. On the basis of the query results evaluated by a utility metric, the learner requests to select most potential instances to be labeled by an oracle (*i.e.* a labeler). After that, the new labeled instances are directly added to the training set D^L to update the model. This process repeats until the model achieves the desired prediction accuracy or the pre-set number of instances are labeled in the training set.

Table 1 Different setting for query strategies

Query strategies	Instance utility		Instance view	
	Uncertainty	Correlation	Feature	Feature-label
IID instance uncertainty	√		√	
Instance correlation	√	√	√	√

2.2 Categorization for active learning methods

The theme of active learning is to select the most informative instances for the current model. Accordingly, "how to select" instances determines how to properly measure each single instance. Several query strategies integrating certain concepts in other machine learning areas, such as information retrieval, are developed to evaluate each instance. A query strategy is chosen to calculate instance utility based on the model prediction result represented by output probability distributions over all possible class labels.

After choosing a certain query strategy, building models to evaluate instances is another issue which needs to be considered. An instance can be queried by a learner or by a committee of heterogeneous learners. Accordingly, the output probability distributions can be computed based on a single model prediction result or a collection of prediction results over all classifier members separately. This corresponds to the second question: "how to evaluate selected unlabeled instances" issue. To summarize, for instance selection for active learning, there are two major research issues: 1) "how to select unlabeled instances for labeling" and 2) "how to evaluate selected unlabeled instances".

2.2.1 How to select unlabeled instances for labeling

We first address the research issue on how to select unlabeled instances for labeling. Two major types of query strategies can be categorized in terms of uncertainty and diversity as shown in Table 1: (1) query strategies based on IID instance uncertainty, and (2) query strategies based on instance correlation, according to the different composition of utility functions. In addition, the instances are also selected from the view of feature space or from feature-label spaces. For traditional single-label learning tasks, the algorithms select the most uncertain sample because the label is unknown, as introduced in Definition 1. In this case, algorithms select instances based on the evaluation from the feature space. For multi-label learning tasks, an instance may have more than one labels, as introduced in Definition 2. Suppose that we are given a portion of the labels for an instance, a method could choose instances based on the evaluation from the feature and the label spaces and consider the feature-label pairs instead of features as an uncertainty evaluation object.

Active learning based on IID instance information is commonly applied in single-label learning tasks, where the utility function is designed based on the input feature space. Moreover, the assumption that the instances in the unlabeled set are treated independently in this scheme makes an uncertainty evaluation function immediately available to calculate the instance utility. Accordingly, methods in this category normally rank instances simply based on the uncertainty metric and choose the ones with the largest uncertainty values for labeling. We categorize query strategies into three groups based on "how to select":

- *Uncertainty Sampling* emphasizes on labeling the most uncertain instances, by using diverse uncertainty schemes such as least confidence, margin, and entropy [21].
- *Expected Gradient Length* focuses on querying instances that cause the maximal change to the current model [46].
- *Variance Reduction* favors instances that minimize the square loss of a learner [1].

Active learning based on instance correlations takes instance correlations into consideration, so the utility metric is a combination of an uncertainty function and a correlation function, whose definition domain can be in sample space or in sample-label space. Based on the different correlation exploration view, *i.e.*, the "how to select" issue, active learning can be further divided into four subgroups:

- *Exploiting on feature correlation*: Usually, a similarity measurement [62] or a correlation matrix [51] on features is utilized to compare pairwise similarities of instances, so the informativeness of an instance is weighted by average similarity on its neighbors. The algorithms rely on clustering algorithms to group instances and select the most representative instances in each cluster to form an optimal subset with maximum uncertainty. This strategy integrates the information density-based metric and the traditional uncertainty measures to evaluate the potential of an instance.
- *Exploiting on label correlation* : Algorithms in this group are widely used in multi-label learning tasks [91], where an instance can have more than one label. In general, the labels have constraints or relations between each other. Therefore, if we are given a portion of labels for an instance, active learning can automatically infer its additional labels using the constraints.
- *Exploiting on both feature and label correlation*: Besides comparison on the feature similarity, this scheme further explores correlations based on the neighbor's prediction information for a specific instance. Therefore, it integrates the result from feature and label dimensions to assess the correlation. This setting is very suitable for mining tasks on an instance set with a complicated structure. However, traditional similarity metrics compute the average similarity over all the pairs of instances, so that the computational cost is expensive as the size of an instance set grows rapidly. Some variance approaches [15,34] are developed following the idea that each cluster has a density center. A simplified version is to compute the similarity between each instance and the center instance by considering the whole set as a cluster and calculate the average of the cumulative result [10].
- *Exploiting on structure correlation* : In this setting, instance correlations are denoted by a weighted graph, where each node represents an instance and each edge represents the relation between two nodes. Intuitively, the nodes with close connection in the graph are likely to have the same label. Following this logics, when one node is annotated, the labels of its neighbors can be inferred, which consequently reduces labeling cost. To achieve this goal, collective classification is a key method used for predicting the labels of nodes in the graph simultaneously.

2.2.2 How to evaluate selected unlabeled instances

Another important issue is how to evaluate instance utility value with the above query strategies. Some algorithms employ a single model, so the instances utility relies on the model prediction result, where the most "ambiguous" instance is the most uncertain one for the model. Others use a set of models to form a "query committee" [43]. In this approach, class label prediction for an instance rests on the majority voting result in the committee. The most

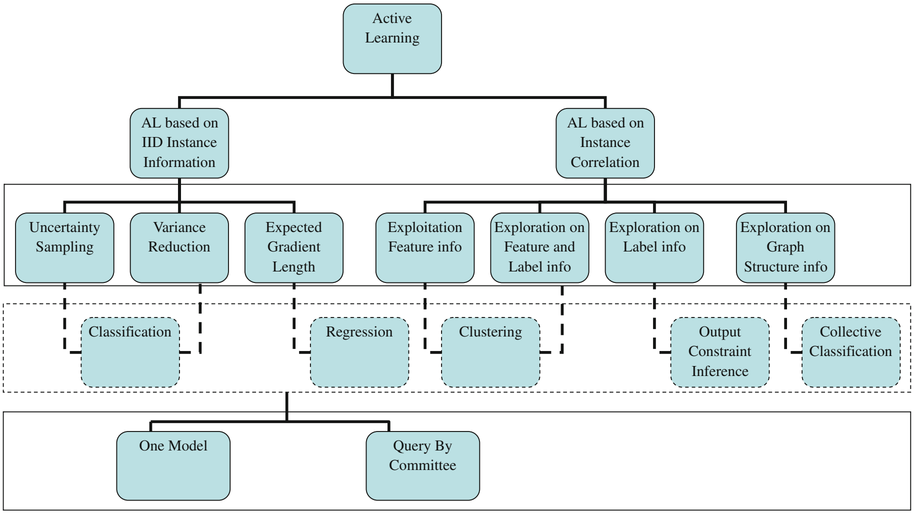

Active Learning
AL based on IID Instance Information
AL based on Instance Correlation
Uncertainty Sampling
Variance Reduction
Expected Gradient Length
Exploitation Feature info
Exploration on Feature and Label info
Exploration on Label info
Exploration on Graph Structure info
Classification
Regression
Clustering
Output Constraint Inference
Collective Classification
One Model
Query By Committee

Fig. 2 Hierarchical structure of the categorization for active learning. The query strategies in the *top solid rectangles* demonstrate “how to select unlabeled instances for labeling”, and the models in the *bottom solid rectangles* demonstrate “how to evaluate selected unlabeled instances”. The methods in the *dash rectangle boxes* explain the implementations of a query strategy

informative instance is the one with the most disagreement prediction from the classifier ensemble.

2.2.3 A combined view

By combining the two research issues, *i.e.*, *how to select* and *how to evaluate*, into a single view, we obtain a hierarchical structure of our categorization for active learning in Fig. 2. It is worth noting that *how to select* can be divided into two major categories and seven subcategories (the solid rectangle on the top), while *how to evaluate* only comprises two categories (the solid rectangle at the bottom). The two dimensions are coupled in certain machine learning tasks such as classification, regression, and clustering (middle dashed rectangle). In the following two sections, we will introduce the two categories of query strategies and their corresponding evaluation methods from an instance-selection perspective.

3 Active learning based on IID instance uncertainty

Definition 7 Active Learning based on IID instance uncertainty: Given an unlabeled sample set D^U, a labeled training set D^L, and an uncertainty function $f_u(.)$ for instance utility evaluation, *i.e.*, $u(.) = f_u(.)$, active learning based on IID instance uncertainty aims to help construct an accurate model by labeling the most informative individual instances in D^L to form the training set D^L, according to $u(.)$.

3.1 How to select unlabeled instances

According to the above definition, the utility of an instance is calculated based on the given uncertainty function, by treating each instance independently. The uncertainty metric is designed based on individual instance importance for an accurate model construction. Many query strategy formulations in active learning use this scheme, and the existing work can

Table 2 Major uncertainty measures for uncertainty sampling

Uncertainty measures	Considered class label	Objective
Least confidence	The one with the highest posterior probability	Decrease the error rate
Sample margin	The first two most probable class labels	Decrease the error rate
Entropy	Over the whole output prediction distributions	Reduce the log-loss

be roughly categorized into three major groups: Uncertainty Sampling, Expected Gradient Length, and Variance Reduction.

3.1.1 Uncertainty sampling

One of the most common frameworks for measuring instances' potential is uncertainty sampling [28], where a component learner regards the most uncertain instance as the ones with the most potential for labeling. This framework often employs probabilistic models to evaluate the potential of instances, such that the prediction result of a single instance is represented by a vector, whose elements are the posterior probability with respect to each class label. Taking a binary classification as an example, the most uncertain instance is the one with 0.5 posterior probabilities with respect to positive and negative class, respectively. According to the number of posterior probabilities considered, uncertainty sampling can be divided into three main subsettings. Table 2 summarizes the main query strategies and their aims in the setting.

A general framework of uncertainty sampling in a multiclass or binary database is the least confidence (LC), developed by Culotta and McCallum [93], with the objective function as follows:

$$x^{\star}_{\mathrm{LC}} = \operatorname*{argmax}_{x} 1 - P_{\Theta}(\hat{y}|x) \tag{5}$$

where $\hat{y}$ is the most likely class label with the highest posterior probability in the hypothesis. This method prefers the instances on which the current hypothesis has the least confidence in deciding their most likely class labels. Based on the basic idea of least confidence, Zhu et al. [63] developed a statistical model for text classification and word sense disambiguation tasks. D-Confidence is proposed for the case of imbalanced class distribution. It selects instances with an improved least confidence criterion, which prefers the instances with low confidence to the known class. Escudeiro and Jorge [13] takes advantage of current classifiers' probability preserving and ordering properties. Li and Ishwar [31] proposed an Active Learning for SVM called "Confidence-based Active Learning" which measures the uncertainty value for each input instance according to its output score from a classifier and selects the instances with uncertainty value above a pre-defined threshold.

In addition to the least confidence which takes an instance's most likely class label into consideration, another popular uncertainty sampling scheme, called the margin approach, integrates the second most probable class label. Margin approach is prone to selecting instances with minimum margin between posterior probabilities of the two most likely class labels [4], which is represented by

$$x^{\star}_{M} = \operatorname*{argmax}_{x} P_{\Theta}(\hat{y_1}|x) - P_{\Theta}(\hat{y_2}|x) \tag{6}$$

where $\hat{y}_1$ and $\hat{y}_2$ are the first and second most probable class labels, respectively. It is easy to understand that the model intends to discriminate between the first and the second most probable class labels. The most informative instances are the ones with the smallest margins between the top two class labels. For large margin machine learning algorithms like Support Vector Machines, the support vectors separate the hyper-plane and maximize the distance between each class. This strategy is generally combined with SVM to construct a more discriminative model. Campbell et al. [6] introduced active learning strategy for support vector selection by building a Support Vector Machine with fewer support vectors. SVM Struct is a flexible resolution for a sequence labeling task. Cheng et al. [11] applies three margin metrics for vector selection under the SVM Struct framework. It considers the input–output combination of a sequence by Conditional Random Field and utilizes dynamic programming to handle the sequence with different lengths. One possible deficiency of the above uncertainty metrics is that they ignore the output distributions for the remaining class labels. Entropy metric is an information-retrieval measure that represents the uncertainty over the whole output prediction distribution. Given a hypothesis Θ, the prediction distribution of an instance x_k, then the uncertainty can be encoded as follows:

$$x_E^{\star} = \underset{x}{\text{argmax}} - \sum_i P_\Theta(\hat{y}_i|x_k) log P_\Theta(\hat{y}_i|x_k) \tag{7}$$

where $\hat{y}_i$ denotes posterior probability of the instance x_k being a member of the i^{th} class, which ranges over all possible labels. For a binary classification task, the most potential instances are the ones with equal posterior probability with respect to all possible classes. Based on the basis entropy theory, various variances of Entropy metrics [5] have been developed in recent decades. As an illustration, N-best sequence entropy metric employs a probabilistic sequence model in natural language processing [26]. Moreover, 'Entropy-Driven Online Active learning', which was developed by Weber and Pollack [55] and was used for interactive calendar management, evaluates the dissimilarity between the schedules at the granularity of a single feature. Holub and Perona [21] proposes an Entropy-based active learning algorithm on a heterogeneous classifier ensemble and applies this strategy to the object category. Entropy metric is extended by accumulating average uncertainty on each feature, respectively, which provides more information than the simple mean over all the features. Recently, exploration of efficient entropy metric via dynamic programming has become a hot topic, and a semi-supervised active learning algorithm is proposed for conditional random fields [33].

According to the above analysis of the three mechanisms, we assert that Entropy metric is appropriate when the objective is to minimize the log-loss function, whereas Margin and Least Confidence metric are more suitable for reducing classification error rate, since they favor the instances helping to discriminate between specific classes.

3.1.2 Expected gradient length

Another conventional query strategy focuses on querying instance that causes the maximal change to the current model. Expected Gradient Length [44] involves querying the instance which could cause the greatest change in the gradient of the objective function if adding it in the training set. The objective function is defined as

$$x_{\text{EGL}}^{\star} = \underset{x}{\text{argmax}} - \sum_{y_i} P(y_i|x;\Theta)\|\nabla\partial(L^{+<x,y_i>};\Theta)\| \tag{8}$$

where $\nabla\partial(L^{+<x,y_i>};\Theta)$ denotes the updated gradient of the log-likelihood ∂ with respect to the new instances x added in the labeled set, and Θ is the model parameter. Because in the query process, the genuine label of the instance x is unknown beforehand, we calculate the summation of log-likelihood over the entire possible label y_i. $\|\ .\ \|$ is the Euclidean norm of each resulting gradient vector and $P(y_i|x;\Theta)$ is the posterior probability of x belonging to class y_i under the model Θ. Since ∂ converges at the previous training set, $\nabla\partial(L,\Theta)$ is nearly zero. Therefore, we use $(L < x, y_i >; \Theta)$ instead of $(L^{+<x,y_i>};\Theta)$ for computation efficiency, because the instances are treated independently in this setting.

Expected Gradient length is widely used in ranking functions, such as web search [29], recommendation, text classification [52]. In many machine learning problems, the quality of a ranking function decides the quantity of labeled instances. Settles et al. [46] adapted this query strategy to multiple-instance active learning. They design two query selection schemes for multiple-instance setting with different granularity in the domain where bag labels are easy to obtain. Long et al. [32] derives a novel metric 'Expected Loss Optimization' to rank unlabeled instances. With the ranking function, a two-stage ELP algorithm is developed by integrating both query and document selection into active learning. However, the computational cost is quite expensive for high-dimensional feature space or a large amount of labeling sets. As a result, the feature space may need to be rescaled in this case.

3.1.3 Variance reduction

Variance Reduction, as its name suggests, intends to minimize the model error rate by selecting instances with the minimum variance. Suppose we are given an instance x, a component learner's expected error, E_T, on x can be decomposed as follows,

$$\begin{aligned} E_T[(\hat{y}-y)^2|x] = {} & E_T[(y-E[y|x])^2] \\ & + (E_L[\hat{y}] - E[y|x])^2 \\ & + E_L[(\hat{y}-E_L[\hat{y}])^2] \end{aligned} \tag{9}$$

where $\hat{y}$ and y are a model's predicted output and a true label for an instance x respectively; $E_L[.]$ denotes an expectation on the labeled set L; $E[.]$is an expectation over the conditional probability $P(y|x)$, and $E_T[.]$ is an expectation over both.

In Eq. (9), the first term on the right-hand side represents noise in a data set. The second term signifies the bias caused by the component algorithm itself, which is stable for a fixed algorithm. The last term is the variance of a learner arouse by ignoring sampling diversity.

Through the above analysis, it is easy to understand that the model has nothing to do with noise and bias; minimizing the variance is the only way to minimize the total expected error. As a result, an algorithm searches the best possible instances to minimize the output variance and the total expected error. Let $\sigma^2_{\hat{y}} = E_L[(\hat{y}-E_L[\hat{y}])^2]$, the Variance Reduction query strategy can be described as follows:

$$x^{\star}_{\mathrm{VR}} = \underset{x}{\operatorname{argmin}} \langle\sigma^2_{\hat{y}}\rangle^{+x} \tag{10}$$

This type of method often takes advantage of a statistical model measuring Fisher Information to evaluate the variance, which is a partial derivative of the log-likelihood function with regard to a model parameter. Minimizing the variance over its parameter estimation is equivalent to maximizing the Fisher Information Function. The advantage of this kind of approach is that the information matrices representing the variance simulate a model retraining process. Settles and Craven [45] extended this algorithm to probabilistic sequence models such as

Conditional Random Fields [18]. However, the biggest challenge is with computational complexity. Each new instance requires a K dimensional square matrix, where K is the number of parameters in a model. The large number of K makes it intractable. A possible resolution is applying a reducing dimensionality technology such as Principle Component Analysis to reduce the parameter space. Hoi et al. [19] formulates batch instances selection into a semi-definite programming problem with a bound optimization algorithm. Hoi et al. [20] solves the Semi-Definitive Programming problem by exploring the properties of submodular function.

In addition to Fisher's information criterion, Vijayakumar et al. [92] took advantage of projection by making a trade-off between expanding the approximation space and reducing variance. Saar-Tsechansky and Provost [42] used bagging sampling to measure the variance in the probability estimates for unlabeled instances. A minimal variance principle is proposed for instances of stream selections, and a dynamic classifier weight updates the global minimal variance [65].

Furthermore, the setting for variance reduction has been applied in the dual control problems, which focus on finding an optimal control law. A lot of solutions are based on the variance minimization in active learning. These approaches [24,35,56] either add a variance term or an innovation process, or consider it as a constraint to perform the active law selection process. However, they always cut the time horizon into several small periods. A new variance-based algorithm taking the global time horizon into account is proposed by Li et al. [30] to find an optimal control law for linear-quadratic stochastic controller problems. Based on this research, it is extended to a discrete time situation in the same problem.

3.2 How to evaluate selected unlabeled instances

Following the above review on "how to select unlabeled instances for labeling", we further explore "how to evaluate utility of the selected unlabeled instances". In general, instances utility is computed based on the model prediction results. According to the number of models constructed in a specific issue, we categorize the evaluation approaches into two types: query by a single model and query by a committee.

3.2.1 Query by a single model

The most straightforward framework is to rely on one single model trained from the training set. After an unlabeled instance is queried, an output probability distribution is generated based on the model prediction result. In this case, the model can choose an effective query strategy to compute the instances importance with respect to the output distributions. By doing so, the most informative instances are the most uncertain ones for the underlying model. Suppose we are given a labeled data set $D^L = \{e^1, e^2, ..., e^s\}$, where s denotes the size of the labeled data set, and e^i denotes the ith instance in D^L. Each instance $e^i = \{f_1^i, f_2^i, ..., f_q^i, y^i\}$ is represented in a feature space $\mathscr{F}$ consisting of each feature value and its class label y. The model construction process is to train a model based on the information provided by D^L. There are numerous methods for single model construction, which can be roughly organized into the following two categories.

Model built based on the whole instance space: This is a basic and simple method for building models. In this scheme, a model is trained on all labeled instances from both feature values and class labels viewpoints. Therefore, the prediction function is an approximate mapping function from the feature space $\mathscr{F}$ to the class label space $\mathscr{Y}$, which can be denoted by

$$p(.) : \mathscr{F} \mapsto \mathscr{Y} \tag{11}$$

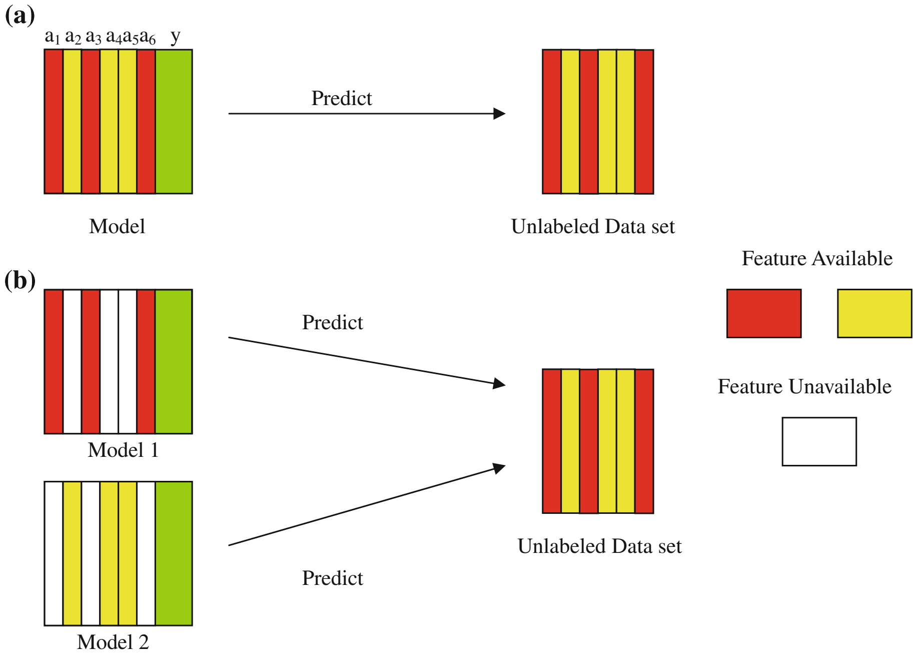


Fig. 3 Difference between **a** model built based on the whole instance space, and **b** model built based on partial instance space. The *red* and *yellow rectangles* denote two independent feature subsets; The *green rectangles* denote label domain; the *white rectangles* represent the features subset is not used in the model (colour figure online)

Model built based on partial instance space: In this case, input feature space is separated into two subspaces ($\mathscr{F} = \mathscr{F}_1 \times \mathscr{F}_2$), where $\mathscr{F}_j$ corresponds to a different view of an instance. Moreover, the two subspaces are conditionally independent. Therefore, each instance e^i is given as a pair of feature subspaces (A_1^i, A_2^i). Assuming that each subspace provides sufficient information for building an accurate model, we can build a model based on information from either "view", whose target function p (.) is denoted by,

$$p(.) : \mathscr{F}_j \mapsto \mathscr{Y}; (j = 1 \; or \; 2) \tag{12}$$

The above model is also commonly applied in co-training classifications tasks, where multiple models, each of which is based on a special "view", are constructed with this strategy. Figure 3 illustrates the difference between the above two model construction methods. Compared with a model built based on the whole input space, this strategy treats the features as conditionally independent, whereas the former treats each feature as an IID. In addition, this strategy reduces the model construction and label prediction costs compared to the former.

3.2.2 Query by a committee

Another well-motivated framework is called *Query By Committee*. This scheme uses a classifier committee constructed from the training set. When an instance is queried, each member makes a vote on the class label of the instance. The final predictions are the majority voting of the committee members. The most informative instance is the one with the most disagreement in the prediction of the classifier ensemble (*i.e.*, committee). The objective of this

Table 3 Major models for query by committee

Model	Sample distribution	Aim
Query by bagging	Not change	Reduce variance
Query by boosting	Change	

setting is to reduce the hypothesis space with a classifier ensemble forming a committee. In other words, it tries to search an optimal hypothesis within the version space, which is a set of models built based on the same training set. *Query by Committee* seeks a few ambiguous instances to find a best model with a small set of hypotheses. In order to implement a *Query by Committee* (QBC) algorithm, two essential tasks need to be completed: (1) construct a committee of hypothesis representing the different fields of a version space; and (2) design a measure to evaluate the disagreements between committee members. The skeleton algorithm is described in Algorithm 2.

Algorithm 2 Algorithm: Query by Bagging

```
Require: The selected optimal subset at each time: $\vartheta$
    The number of classifiers: $\lambda$
    The size of unlabeled instances pool $\mathscr{D}^U$ : $\gamma$
    The initial training set $\mathscr{D}^L$ ;
Ensure: the final classifier model $\hbar$
while do not achieve the budget do
    $S_1, S_2, \ldots, S_\lambda \leftarrow Sampling(\mathscr{D}^L)$;
    $\varepsilon \leftarrow \hbar(S_1), \hbar(S_2), \ldots, \hbar(S_\lambda)$;
    for i=i to $\gamma$ do
        $u_i \leftarrow utility(x_i, \varepsilon)$;
    end for
    $\vartheta \leftarrow$ Select an optimal subset with maximal utility values;
    $D^L \leftarrow D^L \bigcup \vartheta$;
    $D^U \leftarrow D^U \setminus \vartheta$;
    $\hbar \leftarrow \hbar(D^L)$;
end while
```

It is critical for QBC to form a committee with the consistent hypotheses that are very different from each other. Based on different sampling algorithms, two classical practical implementations of this approach are Query by Bagging (*QBBagging*) and Query by Boosting (*QBBoosting*) [82], which use Bagging and Boosting, respectively. We summarize the main idea and the differences between two models in Table 3.

From Table 3, one can conclude that the major difference between *QBBagging* and *QBBoosting* is the building of classifier ensemble. Query by Bagging generates the random instances of the training set using an identical sample distribution, whereas the subinstances are obtained by an input set with changing distributions replying on the hypothesis space. Moreover, the main idea of the two variance approaches of Query by Committee is to reduce the variance of the hypothesis by constructing a set of classifiers on different instances of the same training set.

Numerous algorithms have been developed based on the above two frameworks. Copa et al. [12] presented an unbiased uncertainty measure that is a variance of the Entropy metric under the Query by Bagging framework. It constructs a classifier ensemble on bootstrap instances with bagging technology, so that each classifier member is trained with different parts of the instances in the set, which help achieve the sampling diversity requirement. Moreover, the new approach is tested on image classification tasks with a heterogeneous classifier ensemble, which results in a high convergence rate. At the same time, it has been shown that the original Query by Bagging design does not work well on local classifiers such as the kNN algorithm. This is because the local classifier feature selection is very sensitive to the training process, whereas the classifier ensembles do not focus on selecting useful feature subsets. To address this issue, Shi et al. [48] uses bagging to select the features for local classifiers under an active learning framework.

Conventional Query by Boosting does not have a consistent explicit objective function to combine base learners and query metric. Moreover, the computational complexity for boosting is relatively high, and one cannot dynamically decide the size of the classifier ensemble. To this end, Wang et al. [54] introduced a complicated framework unifying the active learning boosting and semi-supervised learning, building a competitive function integrating the two algorithms. An incremental committee algorithm is developed under the framework, which makes it converge at a low cost. Huang et al. [23] extended the binary Gentle AdaBoost algorithm to a multi-class classification with multi-class response encoding scheme. At the same time, they make use of Query by Committee to query the utility sample. Query by Committee is commonly coupled with different selection strategies, such as least confidence, margin, and entropy and so on. Some strategies take the disagreements between the classifier members into account, while others consider the output class distribution in the classifier ensemble. Zhao et al. [61] developed a new selection scheme by integrating Vote Entropy with Kullback–Leibler divergence under the Query by Committee framework, which choose instances in the terms of class distribution and inconsistency between the classifier members.

QBC seems to reduce the number of labeled instances exponentially. The native algorithm has unreasonable time complexity because of the voting process between hypotheses. Fine et al. [14] applied QBC for linear separation and random walk, involving converting both exponential problems to a convex problem. The key technique is to drop the Bayes assumption if the concept classes have symmetry property. They also set the radius threshold of a ball in a convex body. Many studies have confirmed that QBC can solve linear separation issues with an assumption that hypotheses are presented explicitly under the feature space but it is ineffective for the case of high-dimensional feature space. Gilad-Bachrach and Navor [16] designed a kernel query by committee algorithm, whose running time does not depend on the input dimension, but on the size of labeled data set.

4 Active learning based on instance correlations

Many studies suggest that active learning based on single instance information tends to select outliers [41]. Moreover, the selected instances may also have redundancy. These are because that the uncertainty of a specific instance is calculated based on its own utility with the instance correlation inherently ignored. In order to address this issue, instance correlations are further taken into account to avoid information redundancy.

Definition 8 Active learning based on instance correlations: Given a domain D, consisting of an unlabeled data set D^U and label space $\mathscr{Y}$, an utility function $u(.) = f_u(.) \times q_c(.)$, where $f_u(.)$ is an uncertainty metric function and $q_c(.)$ is a correlation metric function, instance correlation-based active learning tries to select most informative sample/sample-label pairs according to $u(.)$.

Table 4 Different settings of active learning based on instance correlation

Subsettings	Related areas	Instance view	
		Feature	Feature-label pair
Exploration on feature correlation	Clustering correlation matrix analysis	√	
Exploration on label correlation	Multi-label learning constraint inference	√	√
Exploration on feature and label correlation	Clustering	√	
Exploration on graph structure	Collective classification	√	

4.1 How to select unlabeled instances

In instance correlation-based active learning setting, a correlation function is integrated into a utility function with the assumption that instances are dependent on each other. Compared with Definition 4, this strategy selects instances from both correlation and uncertainty views rather than from the uncertainty aspect only. For some multi-label tasks, we are given a portion of labels of an instance, so a sample-label pair can be considered as an object to assess its utility instead of using feature values of the instance only. Based on different types of correlations, we categorize this setting into three subcategories as shown in Table 4 and summarize the relationship between traditional machine learning and various subsettings in this strategy.

4.1.1 Exploration on feature correlation

In this scheme, most algorithms exploit feature correlations by using clustering methods. Many studies have shown that semi-supervising algorithms can facilitate the clustering process. In active learning setting, a small labeled training set and a large unlabeled pool provide a semi-supervised learning environment, which makes it possible to incorporate clustering algorithms to group instances before an active query starts. A simple clustering partitions instances based on features information. In the clustering algorithms, a similarity measure is developed as grouping criteria. Thus, the utility of a single instance is weighted by the average similarity over all other instances in the unlabeled set, which is described as follows:

$$x^\star = \underset{x}{\operatorname{argmax}}\, f_u(x) \times \left(\frac{1}{U}\sum_{u=1}^{U} \mathrm{sim}(x, x^u)\right)^\beta \tag{13}$$

where $f_u(.)$ is the uncertainty function of a single instance, which is calculated according to the definitions given in Sect. 3.1, U denotes the size of unlabeled data set, $sim(x, .)$ denotes the similarity function to evaluate the distance between two instances, and β controls the importance of density term. The utility of an individual instance is weighted by average similarity over the unlabeled data set. The most representative instances selected from each group form an optimal subset with maximum uncertainty.

Traditional similarity metrics compute the average similarity for each instance. For data set with large volumes, the computational cost can be expensive. Some alternative approaches [15,34] have been developed based on the idea that each cluster can be considered as a dense

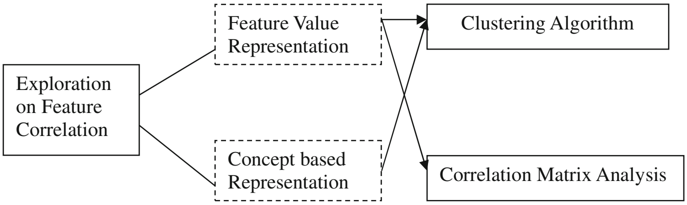

Exploration on Feature Correlation
Feature Value Representation
Concept based Representation
Clustering Algorithm
Correlation Matrix Analysis

Fig. 4 Relationships between the algorithms used in exploration on feature correlation and evaluation granularity

region in the input space. Chen and Subramani [10] implemented a simplified version of the information density function by computing the similarity between the instances and the mean point. This function requires less computation than Eq. (13) under assumption that there is only one cluster in the data set. Depending on different corpus used in real-world applications, the first term of Eq. (13) can be replaced by other uncertain sampling schemes, such as least confidence, margin and so on. In addition to the exploration on features correlation using clustering algorithms, Sun [51] utilizes a feature correlation matrix to measure the property difference between pairwise examples.

Furthermore, the above approaches for feature correlation fall into two cases based on "how to evaluate selected unlabeled instances". Traditional algorithms represent instances using an feature vector, so that similarity function evaluates instance correlation in terms of feature values. In fact, correlation exploration on feature value is insufficient. Take text classification as an example, conventionally, the similarity comparison on pairwise documents is measured by accumulating the occurrence rate of each term/word. This approach can only group documents with many identical words, while semantic relations like acronyms and synonyms are ignored. To this end, the *content (or semantic)-based similarity* measure has been proposed. Huang et al. [22] utilized Wikipedia to design a concept-based representation for a text document, and evaluated the content correlation of pairwise documents at the granularity of Wikipedia concepts to find instance-level constraints. Nguyen and Smeulders [37] proposed a representative active learning algorithm considering the prior distribution of the instance set. However, one weakness of this method is that it is only effective in linear logistic regression, which means all the clusters are modeled with the same parameters. To address this issue, Yan et al. [58] introduced a new framework for semi-automatic annotation of home video under the active learning strategy. In their design, an off-line model is built on a small labeled data set. The initial model is adjusted with the local information of a specific home video obtained online. In this paper, they used four semantic concepts to present the labeling process.

The relationships between the algorithms used in this strategy and evaluation granularity are summarized in Fig. 4.

In the previous work, various similarity measures are designed for feature correlation exploration. There are three commonly used similarity functions.

A) Cosine Similarity Cosine similarity is a measure which calculates similarity between two vectors by finding the cosine of the angle between them. The formula is defined as follows:

$$\mathrm{sim}_{\cos}(x, x^u) = \frac{\overrightarrow{x} \cdot \overrightarrow{x_u}}{\parallel \overrightarrow{x} \parallel \times \parallel \overrightarrow{x_u} \parallel} \tag{14}$$

where $\overrightarrow{x}$ is a fixed-length feature vector of the instances x, representing the inner dot product of two vectors, and $\parallel \cdot \parallel$ is the vector norm.

Cosine similarity is widely used in sequence instance classification, especially in sequence classification tasks. For instance, an improved k-Nearest Neighbor Algorithm-based Cosine Similarity is applied for text classification [29]. A multi-criteria-based active learning for name entity recognition exploits the utility of an instance from three criteria, including formativeness, representativeness, and diversity [47]. Meanwhile, it employs two criteria combinations to select instances. Cosine similarity is also used to measure similarity between words in the representativeness strategy.

Cosine similarity is very effective for instances with low-dimensional features, which avoids unnecessary computation. It evaluates the similarity on the original input space without subspace transition or matrix connection. Nguyen and Li [38] proposed a cosine similarity metric for face verification. Unlike comparing two face based on the traditional Euclidean distance metric on a transformed subspace, they used cosine similarity on the original input space. This similarity metric also plays an important role on text-independent speaker verification. Classical variable score normalization techniques define speaker subspace and channel factors separately, and estimate them jointly. To reduce computing complexity, Shum et al. [49] proposed a new score normalization scheme with cosine similarity, effectively reducing the additional computation at each adaptation update process.

B) KL Divergence Similarity Kullback–Leibler divergence is a non-symmetric measure capturing difference between two instances. The utility of each instance is weighted by the summation of the difference over the rest instances in the unlabeled pool. The exponential KL divergence similarity function is defined as

$$\mathrm{sim}_{\mathrm{KL}}(x, x^u) = \exp\left(-\gamma_1 \sum_{j=1}^{J} P(f_j|\overrightarrow{x})\log\frac{P(f_j|\overrightarrow{x})}{\gamma_2 P(f_j|\overrightarrow{x_u}) + (1-\gamma_2)P(f_j)}\right) \tag{15}$$

The smoothing parameters y_1 and y_2 control the divergence speed and encoded distribution in the denominator, respectively. $\overrightarrow{x_u}$ is a J-dimensional feature space, $P(f_j|\overrightarrow{x})$ denotes posterior probability of containing a feature f_j. $P(f_j)$ is simply a marginal probability of feature f_j over all instances in the pool.

KL divergence has been applied to active learning to evaluate different class output distribution between the classifiers in the ensemble, such as named entity recognition [2] and information extraction [25]. Because KL divergence has a non-negative value, the larger the value, the more different the pair is, and a zero KL divergence value indicates two identical distributions. When taking a peaked distribution as a benchmark of certainty, KL divergence is very similar to cost-testing [36].

Due to the non-symmetric property of the measure, the similarity between pairwise instances should be computed twice, implying a high computational cost. In order to improve the computational complexity, Zhao et al. [60] developed an active learning model based on this strategy for telecom client credit risk prediction.

C) Gaussian Similarity Another exponential similarity measure used to estimate information density is called Gaussian Similarity, which is an exponential Euclidean distance aggregating all the distances on each feature. The formula is represented as

$$\mathrm{sim}_{\mathrm{Gauss}}(x, x^u) = \exp\left(-\sum_{j=1}^{J}\frac{(\overrightarrow{x}^{j} - \overrightarrow{x_u}^{j})^2}{\alpha^2}\right) \tag{16}$$

where α^2 is the variance in the Gaussian distribution. Different variance can be set for different feature, but there is a challenge for setting appropriate parameters. Moreover, it has been suggested that a model with several parameters does not improve the effort of representing the similarity. A semi-supervised learning using Gaussian fields and harmonic functions integrates this query scheme to select most informative instances [64]. Zhu et al. [57] further introduces a combination of active learning and semi-supervised learning under the above framework. Moreover, in [27], graph kernel functions based on the inverted square Euclidean distance and Gaussian similarity, respectively, are evaluated in the context of rating prediction problems. The experimental results show that Gaussian functions outperforms other kernel functions in most cases.

4.1.2 Exploration on label correlation

For multi-label and multi-task problems, the output space contains multiple class related labels, which means that outputs are subject to some inherent correlations, such as agreement, inheritance, exclusive and so on. These correlations provide valuable information for reducing prediction cost. To this end, many studies have leveraged output constraints to improve the learning process [7–9], where the label correlation is used for model parameter estimation or inference on unlabeled instances. Zhang [59] introduces a cross-task information metric for multi-task active learning, whose utility is measured by all the relevant tasks reachable through task output constraints. This framework combines uncertainty sampling metric with inconsistency of prediction on coupled tasks. The main procedures of the algorithm are as follows.

(1) *Choose a reward function for a single task*: Given an unlabeled sample x, a utility function UI is used to compute its importance for improving model performance, which can be denoted by

$$\mathrm{UI}(Y, x) = \sum_{y} \hat{p}(Y = y|x) R(\hat{p}, Y = y, x) \tag{17}$$

where $\hat{p}$ represents the posterior probability of sample x belonging class y, $R()$ is a regard function. This formula accumulates the reward on each possible label y.

(2) *Specify the constraint set between task outputs* By constructing the propagation rules, the algorithm computes the set of propagated outcomes for each possible label. The set of propagated outcomes $\mathrm{Propc}(Y_i = y_i)$ is defined as the inferred outcome labels from task $Y_i = y_i$ based on constraint.

$$\mathrm{Propc}(Y_i = y_i) = \{Y_j = y_j | Y_i = y_i \underset{C}{\rightarrow} Y_j = y_j\} \tag{18}$$

Based on the propagated outcomes, the reward function $R(Y_i = y_i, x)$ is defined as follows:

$$R(Y_i = y_i, x) = \sum_{\substack{Y_j = y_j \in \mathrm{Prop}_C(Y_i = y_i) \\ Y_j \in UL(x)}} R(\hat{p_j}, Y_j = y_j, x) \tag{19}$$

(3) *Compute cross-task value of information for a sample-task pair*: With Eq. (19), the cross-task utility for a sample-task pair is defined as follows.

$$\mathrm{UI}(Y_i, x) = \sum_{y_i} \hat{p}_i(Y_i = y_i|x) \sum_{\substack{Y_j = y_j \in \mathrm{Prop}_C(Y_i = y_i) \\ Y_j \in \mathrm{UL}(x)}} R(\hat{p_j}, Y_j = y_j, x) \tag{20}$$

With the above three steps, the algorithm selects the most informative sample-label pair which maximizes the UI value.

Qi et al. [40] proposed a two dimensional active learning algorithm for multi-label problems, which explores the uncertainty of sample and label correlation concurrently. The novel method requests the annotation on the sample-label pair, once added into the training set, is expected to minimize generalization error. In their paper, they derived a Multi-labeled Bayesian Error Bound for the sample-pair selection.

Given a sample x and its labeled and unlabeled parts $U(x)$ and $L(x)$. Once y_s is activated to ask for labeling, the Bayesian classification error $\varepsilon(y|y_s; y_L(x), x)$ for an unlabeled $y_i \in U(x)$ is bounded as:

$$\varepsilon(y|y_s; y_L(x), x) \leq \frac{1}{2m}\sum_{i=1}^{m}\{H(y_i|y_{L(x)}, x) - MI(y_i; y_s|y_{L(x)}, x)\} \tag{21}$$

where

$$H(y_i; y_s|y_{L(x)}, x) = \sum_{t,r\in\{0,1\}}\{-P(y_i = t, y_s = r|y_{L(x)}, x)\log P(y_i = t, y_s = r|y_{L(x)})\} \tag{22}$$

which denotes the entropy of the sample-label pair and $MI(X; Y) = H(X) - H(X|Y)$, denoting the mutual information between y_i and y_s given unlabeled part $U(x)$. Accordingly, the algorithm selects the most informative sample-label pairs, which are expected to reduce the Bayesian classification error over the unlabeled pool to the greatest extent. Before selecting a sample-label pair (x_s, y_s), the expected Bayesian classification error is denoted by

$$\epsilon^b(P) = \frac{1}{|P|}\sum_{x\in P}\epsilon(y|y_{L(x)}, x) \tag{23}$$

After the pair is selected, the expected error is calculated as follows

$$\epsilon^a(P) = \frac{1}{|P|}\left\{\epsilon(y|y_s; y_{L(x)}, x_s) + \left(\sum_{x\in P-x_s}\epsilon(y|y_{L(x)}, x)\right)\right\} \tag{24}$$

Therefore, the goal of the algorithm is to select a best (x_s^*, y_s^*) which maximizes the error reduction $\triangle\varepsilon(P)$, that is

$$\begin{aligned}(x_s^\star, y_s^\star) &= \arg\max_{x_s\in P, y_s\in U(x_s)} \triangle\varepsilon(P)\\ &= \arg\min_{x_s\in P} -\triangle\varepsilon(P)\end{aligned} \tag{25}$$

where $\triangle\varepsilon(P) = \varepsilon^b(P) - \varepsilon^a(P)$. Applying Eqs. (22–25), we have

$$(x_s^\star, y_s^\star) = \arg\max_{x_s\in P, y_s\in U(x_s)} MI(y_i; y_s|y_{L(x_s), x_s)} \tag{26}$$

Consequently, the method selects the sample-label pair for labeling according to Eq. (26), by maximizing the mutual information at each loop.

4.1.3 Exploration on feature and label correlation

Some algorithms exploit both feature and label correlations simultaneously. In addition to comparison on the feature similarity, this scheme further explores the neighbor's prediction information for a specific instance. Godec et al. [17] designs a hidden multi-class representation to capture intra-class variability for object detection for an image, i.e., a binary classification task discriminating foreground from background. In their design, they applied a classifier-based bootstrapping with online multi-class classifier and generated virtual classes, which are separated into negative and positive classes. Based on these label correlations, each modality decides whether to generate a new virtual class. Then we utilize clustering to find a context background, which explores sample correlation from the view of features. The main process is as follows: In the first step, it trains an initial classifier to discriminate object from background, and then applies it to the current scene. For each sample, misclassified by the model or close to the decision boundary, a new virtual class is added to the multi-class models. In this model, Gradient Boost algorithm is used to combine a number of selector f_m, to a strong one

$$F(x) = \sum_{m=1}^{M} f_m(x) \tag{27}$$

Each selector f_m is formed by a number of classifiers $\{f_{m,1}(x), \ldots, f_{m,N}(x)\}$, and is represented by its best classifier which minimizes the generalization error. For each $f_{m,i}(x)$, online histogram is used to evaluate their confidence on prediction. For example, one can use symmetric multiple logistic transformation in Eq. 27, Where p_j can be calculated in the online histogram. Since the model is built on the context of scene, it can also handle changing context.

$$f_j(x) = \log p_j(x) - \frac{1}{J} \sum_{k=1}^{J} \log p_k(x) \tag{28}$$

4.1.4 Exploration on structural correlation

A graph is a good data structure to present instance correlation in a data set. Given a graph $G = (V, E)$, each node V_i denotes an instance x_i, which is denoted by a vector as introduced in Definition 1. Each edge $E_{ij} =< V_i, V_j >$ describes some relationship between two nodes V_i and V_j. Take web page link as an example, the web page is likely to have similar topics with its link pages; therefore, each web page is denoted by a node, and the link relationship between the web pages is denoted by the edge. Consequently, we can explore instances correlation from the graph structure. When the label of a node is annotated, the labels of its neighbors can also be inferred, which reduces labeling cost.

In this graph structure setting, collective classification is a key method used for predicting labels of nodes in the graph simultaneously. Generally, the label y_i of a node x_i depends on its own features as well as the labels y_j and features of other nodes x_j.

Various collective classification methods have been proposed with regard to the structural correlations. Bilgic et al. [3] proposed a novel active learning for network instances. They constructed a local collective classification model on a complex object, which is a vector including its local features x_i and an aggregation of features and labels of its neighbors aggr(N_i). The collective classifier CC is used to learn $P(y_i|x_i, \text{aggr}(N_i))$, and a content-only classifier CO based on local node information is used to learn $P(y_i|x_i)$. After clustering the nodes, the algorithm calculates the disagreement score of each cluster C_j, which is defined as

$$\text{Disagreement}(CC, CO, C_j, D^L) = \sum_{V_i \in C_j \cap D^U} LD(CC, CO, V_i, D^L) \tag{29}$$

where $LD(CC, CO, V_i, D^L)$ is the entropy value of node V_i's labels over the output spaces.

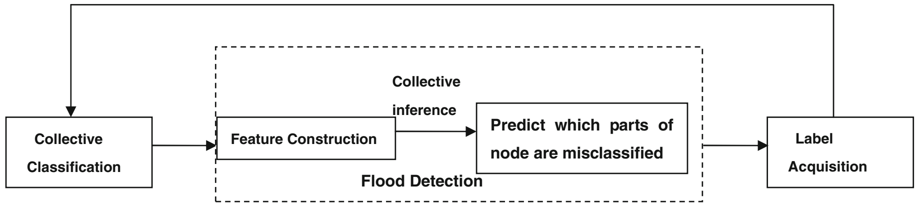


Fig. 5 Active inference using Reflect and Correct Method. We iteratively label the nodes with a collective model. To predict node mis-classification, we build a classifier on the features containing node content and its neighbor information, and then predict possible mis-classified instances with the classifier, tagging their mis-classified label as True. After that, they are corrected with RAC strategy

After computing the disagreement score of each cluster, one ranks them according to their disagreement score and select the first k clusters with the largest score, and randomly sample an item in each cluster for labeling.

Notice that when a given collective classification misclassify a node, an island of nodes is likely to be misclassified. To address this issue, Bilgic and Getoor [66] added a "reflect and correct" scheme under a collective classification model based on their previous work. They developed an active inference for collective classification, with general process illustrating in Fig. 5.

The method constructs a traditional collective model on the graph, and then make predictions on the test instances. To find out whether a node is misclassified or not, it constructs a feature table that is a possible indicator for judging whether a node is misclassified and builds a classifier on the features to predict the possible misclassified ones. After that, it acquires a label for the central node among the potentially misclassified ones. The process repeats until the system's requirements are satisfied. This iterative process is called reflect and correct process.

4.2 How to evaluate selected unlabeled instances

After exploiting instances correlation from four different views, we utilize the same framework introduced in Sect. 3.2 to study the evaluation of selected unlabeled instances.

Indeed, there are some extra model construction methods for instance correlation-based active learning algorithms. For Query by Single model, the application has been expanded into multi-label tasks. Suppose given a labeled data set D^L, each instance e^i is an multi-label instance as denoted in Definition 4, which is denoted as $x^i = \{f_1^i, f_2^i, \ldots, f_q^i, y_1^i, \ldots, y_m^i\}$, and label constraints rules $\mathscr{C}$. However, an instance may have incomplete label information, that is, only a portion of labels $\mathscr{Y}^L$. With such incomplete information, an accurate model still can be built by taking advantage of label constraints. To this end, a new model based on feature information and label constraints is developed for multi-label prediction tasks. A model can be constructed according to feature information $\mathscr{F}$ and known label information $\mathscr{Y}^L$, because the unknown labels $\mathscr{Y}^U$ can be inferred with constraint rules $\mathscr{C}$, which effectively reduces the labeling cost. Therefore, the target function is represented as follows:

$$p(.) : (\mathscr{F}, \mathscr{Y}^L) \underset{\mathscr{C}}{\rightarrow} \mathscr{Y}^U \tag{30}$$

While for the Query by Committee model, algorithms exploring graph structure correlations have a different committee construction method. They build two classifiers to form a classifier committee, including a collective classifier and a context-only classifier. The

Table 5 Summary of all reviewed instance-selection methods in terms of two dimensions: “how to select” and “how to evaluate”

How to select			How to evaluate	
			One model	Committee
IID Instance uncertainty	Uncertainty sampling [UNG] [28]	Least confidence [UNGLC]	[31,63]	[68]
		Margin [UNGMA]	[6,11]	[18]
		Entropy [UNGEN]	[5,26,33,87]	[12,21,61]
	Expected gradient length [EGL]		[29,46]	[32]
	Variance reduction [VAR]		[18,19,24,45]	[30,35,42,56,65]
Instance correlation	Exploiting feature correlation [EFC]	Cosine similarity [EFCCS]	[29,38,47]	[49]
		KL divergence [EFCKL]	[2,25]	[36,60]
		Gaussian similarity [EFCGS]	[27,57,64]	
	Exploiting label correlation [ELC]		[7–9]	[40,59]
	Exploiting feature and label correlation [EFLC]		[17]	
	Exploiting structure correlation [ESC]			[3]

former predicts an instance class label according to its own feature information, as well as its neighbor’s feature and labels information; whereas the latter is built based on its own information. Query by Committee favors the instance maximizing the disagreement between the two classifiers. Compared with the committee used in the active learning based on IID information, the committee member consists of different kinds of classifiers rather than the same types of classifiers.

5 Algorithm performance comparison

In this section, we first summarize all the reviewed instance-selection methods in Table 5, with respect to the two dimensions: “how to select” and “how to evaluate”. Then, we select some representative methods from the two major categories: active learning based on IID instance uncertainty and active learning based on instance correlations, to conduct an experimental study and compare their performance, as well as analyze their strengths and weakness.

5.1 Performance analysis

Most papers have shown that active learning gains improvements compared to passive learning. In the literature, since algorithms are tested and evaluated in different experimental settings, it is difficult to make a fair comparison across various active learning methods. In this subsection, we focus on the computational time complexity of some representative algorithms in each category introduced above. To simplify the representation, the algorithms used in the section are represented by the abbreviation in Table 5. Because the time complexity for various algorithms relies on the component learner used in a method, which has a different computation complexity, we cannot make fair comparisons. In this paper, we evaluate the time complexity of different algorithms based on the time cost for a query process over the unlabeled data set. We simply summarize the above query strategies in

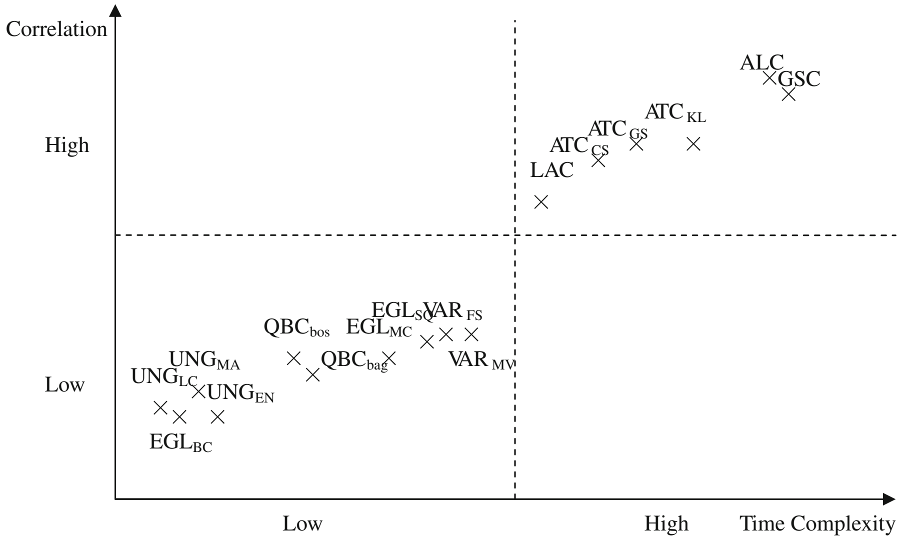


Fig. 6 Query strategy comparison on representative algorithms from the instance correlation and time complexity perspectives. The x-axis denotes the time complexity and the y-axis denotes the instance correlations

Table 5 from the dimensions of time complexity and instance correlation, as shown in Fig. 6. From Fig. 6, the time complexity and correlation values of the algorithms taking instance correlation into account are much higher than the ones based on IID information, which suggests algorithms taking correlation into consideration require more time to explore correlation information. For active learning based on IID information, we can easily conclude that Uncertainty Sampling strategies have the lowest time complexity, whereas Variance Reduction and Expected Gradient length algorithms have a higher time cost among the four strategies. The observations suggest that simple query strategy costs less time than complex strategies. For instance, Fisher algorithms need a K dimensional matrix in the calculation, whereas uncertainty sampling just uses the output distribution to evaluate instances uncertainty. The more details considered in the algorithm, the higher time complexity it requires. For Expected Gradient length scheme, its application on binary classification has the comparable performance with Uncertainty Sampling. However, for the multi-class prediction and sequence mining tasks, the time complexity becomes higher as the number of class labels or the length of sequence grows. The performance of Query By Committee is superior to EGL and VR, but is inferior to UNG, the main time cost depend on the component learner it chooses.

When taking instance correlation into consideration, the algorithms exploiting both feature and output information have almost the same time complexity as the ones exploiting correlation based on graph structure. The algorithms mining feature information with clustering algorithms have second highest time complexity. The above observations are consistent with our expectation. The more information explored, the more time the algorithm needs. For feature based algorithms, the method with KL divergence has a relative higher time cost than the other two similarity metrics. This is because KL divergence is an asymmetric metric, which costs more in computation cost. For output relation exploited algorithms, the time complexity relies on the number of class labels, which is much less than the number of data sets. Therefore, these kinds of algorithms have lower time complexity than the algorithms based feature information.

5.2 Lessons learned

5.2.1 Lessons from IID-based active learning

IID-based active learning employs an uncertainty evaluation measure to calculate instance utility values by treating instances as IID samples. All unlabeled instances are ranked based on their uncertainty values, and the subset with the largest uncertainty values are selected for labeling.

IID based approaches are commonly used in single-label learning tasks. The three representative subgroups are suitable for different types of applications.

- *Uncertainty sampling* [13,21,93] is often straightforward for probabilistic learning models. For example, *least confidence* has been popular with statistical sequence model in information extraction [45,93]. In addition, *entropy*, a general uncertainty sampling measure, is appropriate to minimize the log-loss of the objective function [31,63]. The other two uncertainty sampling measures, *least confidence & margin*, are more suitable for reducing model error [4–6,33,55], because they favor instances helping to discriminate specific classes.
- *Expected Gradient Length* is widely used in applications involving ranking functions [29,52], such as information retrieval and text classification. Moreover, *Expected Gradient Length* strategy can be applied to discriminative probabilistic models by using gradient-based optimization, where the "change" of the model is evaluated by the length of the training gradient [32,46].
- *Variance Reduction* can avoid model retraining process by taking advantage of Fisher Information Function: the information matrices simulate retraining process with an approximation of output variance [18–20,45]. The setting for variance reduction has been applied in the dual control problems as well. These approaches [24,35,56] either add a variance term or an innovation process, or consider it as a constraint to perform the active law selection process.

When taking individual instance uncertainty value into consideration, the selected instance subset may contain redundant knowledge and therefore cannot form an ideal candidate set. In addition, for *Expected Gradient Length* and *Variance Reduction* based methods, there are some practical disadvantages in terms of computational complexity. For high-dimensional feature space or large data sets, *Expected Gradient Length* is computationally expensive and its performance can deteriorate significantly if features are not appropriately scaled. In other words, the instance utility value calculated by expected gradient length can be over-estimated simply as a result of either one or multiple feature values or the corresponding parameter estimation is quite large, both resulting in a gradient of high magnitude. Meanwhile, the biggest challenge of *Variance Reduction* is its computational complexity. Each new instance requires a $K \times K$ matrix inversion for output variance estimation, where K is the number of model parameters, resulting in a time complexity of $O(UK^3)$ (U denoting the size of unlabeled instances). Therefore, for complex models involving a large number of parameters (K), the computational complexity of variance reduction-based approaches can be very large. As a result, *Expected Gradient Length* and *Variance Reduction* are empirically much slower than simple uncertainty measuring strategy like *Uncertainty sampling*.

5.2.2 Lessons from instance correlation-based active learning

Comparing with *IID-Based Active Learning*, instance correlation-based active learning explores relationship between instances to calculate utility values of unlabeled samples. An utility metric is a combination of both an uncertainty function and a correlation function. Therefore, the selected candidate set balances the instance uncertainty and diversity for active learning. According to the different correlation exploration views, existing solutions in this category are further be categorized into four groups: *Exploiting on feature correlation*, *Exploiting on label correlation*, *Exploiting on both feature and label correlation* and *Exploiting on structure correlation*.

Different from IID-based active learning which is mainly used for single-label learning tasks, instance correlation-based active learning has been used for single-label [10,34], multiple-label tasks [59], and for data with complex structures [17].

- *Exploiting on feature correlation* is the most common way to calculate instance correlations through feature-based similarity measurements. Among all types of similarity measures, *Cosine Similarity* is adequate for sequence classification tasks, such as text classification [29] and name entity recognition [47]. Meanwhile, Cosine similarity is very effective for instances with high-dimensional features, such as face recognition and text classification [38,49], because it evaluates the similarity on the original input space without subspace transition or matrix connection. *KL Divergence Similarity* is very unique and useful for evaluating the similarity between class distributions generated from different classifiers [2,36]. *Gaussian Similarity* works well in semi-supervised learning frameworks [57,64] and graph kennel function [27].
- *Exploration on Label Correlation* aims at solving multi-label learning and multi-task learning problems by exploring the output constraints to improve the learning process [7–9], as well as to reduce the prediction cost.
- *Exploiting on both feature and label correlation* mainly handles data set with multiple labels by considering feature and label correlations at the same time. For example, one can capture multi-class correlation to represent intra-class variability for visual object detection and tracking [17].
- *Exploiting on structure correlation* denotes instance correlation using a graph representation, assuming that an instance' neighbors share the same labels as the instance. Collective classification is a key method employed for predicting the labels of nodes in the graph simultaneously. This setting is applicable to networked data [3] and for active inference problems [66].

While instance correlation-based active learning is effective to reduce redundancy in the selected candidate set, the computational cost for instance correlation calculation is expensive, especially for data sets with a large number of instances. For non-symmetric similarity measures, such as *KL Divergence Similarity*, the computation cost is twice high as the symmetric measures. The parameter settings, especially for *Gaussian Similarity*, can also be a big challenge. In addition, a common assumption in *Exploiting on structure correlation* is that an instance's neighbor nodes share the same labels as the instance. In reality, collecting enough labeled nodes are difficult (or impossible), so that prediction results mainly depend on the network structures, which may reduce the prediction accuracy. Meanwhile, clustering is often the first step for collective classification, where the quality of the clustering results can bring big impact to the final results and result in sampling redundancy in the selected candidate set.

6 Emerging applications: challenges and trends

The methods reviewed in the previous sections are all in the standard active learning setting, in the senses that both labeled and unlabeled data sets are available before training, samples are assumed to be IID in the feature space, and labels are assumed to be provided by domain experts. However, in many emerging applications, these conditions can hardly be satisfied. Many challenges are posed for active learning in various complicated scenarios. The instance-selection methods for active learning in these complicated scenarios are urgent to explore. In the following, we summarize a number of emerging active learning scenarios. For each scenario, we will analyze their challenges and discuss the research trends.

6.1 Active learning on streaming data platform

In many real-world applications, a large number of unlabeled instances arrive in a streaming manner, making it difficult (or even impossible) to maintain all the data as a candidate pool, such as email spam detection, malicious/ normal webpage classification [97]. This type of application face two issues. First, it generates diverse and massive data volumes in a short period of time, making it impractical for domain experts manually examining every datum. Second, the data stream evolve over time; therefore, the traditional training methods on a static data set may fail. A natural solution to tackle the two issues is to employ active learning by selecting a small amount of informative data for labeling to help build a model. However, traditional active learning does not fit for the dynamically changing candidate pool. Thus, the challenges of active learning on streaming data platform is threefold: (1) In the streaming data platform, the data volumes come continually, the candidate pool is dynamically changing, which also leads to the data distribution and decision boundary is evolving consecutively, whereas traditional active learning can deal with only static data sets. (2) Because of increasing data stream, storing all the data is very costly and impossible, whereas traditional active learning use a candidate pool to store all the data in a data set. (3) In the data stream framework, because of the drifting/ evolving data volumes, building a model based on all the labeled data may not be reasonable, while traditional active learning rely on a model build from all the previously labeled data.

To tackle these challenges, several algorithms have been proposed recently [67,83,84,95, 96]. A Minimal Variance principle [65] is introduced to guide instance selection from data stream, coupled with a dynamic weight updating rule for data stream with drifting/evolving concepts. Following the same principle, Chu et al. [67] considered unbiased property in the sampling process in data streams, design optimal instrumental distributions in the context of online active learning. Zhang et al. [68] presents a weighted ensemble classifiers and clusters model to mine concept drifting data streams. Most existing work on streaming data platform rely on building accurate ensemble models [83]. They share the same basic idea: using divided-and-conquer techniques to handle large volumes of data stream with concept drifting. Specifically, the data stream is partitioned into several small chunks, with each ensemble member model constructed from one chunk. The member models are eventually combined in different ways for prediction.

6.2 Active learning with complex data presentations

The massive data set collections of networked data in various domain applications (*e.g.*, social network, information network, and document citation) drive the research of flexible and accurate graph-based prediction models. Vertices classification in a graph is an important topic in graph-based models. A simple graph is designed based on a relation measure (*e.g.*, distance or similarity), where each node denotes a data element, and the edge denotes the relation between the corresponding pair of nodes. Given a graph with unlabeled vertices and a subset of labeled vertices, a model infers the unlabeled nodes memberships, on the strength of labeled training set and graph structure. A common assumption which a model depends on for vertices classification is that similar data should be assigned the same class label. Thus, unlabeled vertices are given the label of its nearest labeled neighbor in a simple way. However, in most cases, gathering sufficient labeled vertices are very expensive, so prediction results depend mainly on network structure, which may reduce performance. To address this issue, active learning focus on reducing labeling cost by selecting an optimal utility vertices in a graph for the purpose of constructing a superior model.

In general, existing vertices selection criteria falls into two categories. The first type of approaches is to find an optimal solution for a designed objective function. For instance, [70] proposes a function which seeks an optimal labeling vertices V^L that disconnect most regions of the graph by cutting minimal edges. However, there is no general algorithms for minimizing the function, and the method may perform worse than random algorithms in some experiments [70]. To address this issue, Cesa-Bianchi et al. [69] employs active learning algorithm to find the minimization of the objective function on a spanning trees. Unfortunately, there is no experiments showing its effectiveness on a general graph. They conduct experiments with random spanning tree (RST) and breadth-first spanning tree (BST). RST may hide the cluster structure of graph, while, BST are likely affected by parameters like starting node.

Another kind of algorithms selects vertices corresponding to the disagreement between classifiers. In their designs, they employ clustering algorithm to group vertices based on graph topology. Then, they make predictions on each cluster by using a classifier community, and select samples with the most disagreement in each cluster to form an optimal subset. Bilgic et al. [3] effectively exploits the prediction difference between a classifier and a collective classifier, where the former is built with vertices information, while the latter also takes edges between vertices and neighbor's information into consideration. However, a fixed number of clusters are likely to destroy the actual data class distribution.

6.3 Active learning with crowdsourcing labelers

Traditional active learning asks an omniscient expert to provide ground truths to the queried instances, so that labeled instances can help build an accurate model. By doing so, the expert is assumed to be accurate (never wrong), indefatigable (always answers the queries), unique (only one oracle), and insensitive to costs(inexpensive/free annotation cost). However, labeling an optimal utility subset is still costly and expensive in many cases. To reduce labeling cost, crowdsourcing labelers, which are composed of some cheap and noisy labelers, have now been considered for active learning. Unfortunately, a direct application of crowdsourcing labelers on traditional active learning is problematic for two reasons. (1) Since only a small subset of critical instances are selected for labeling, the labeling quality in active learning is more sensitive to the model's performance. (2) Since active learning is consisted of multiple learning iterations, the errors induced in each round will be passed onto the following rounds and will be amplified. Thus, asking crowdsourcing labelers to directly provide noisy class labels may not be appropriate in active learning. The tradeoff between the labeling noise and labeling cost is a big challenge for active learning with crowdsourcing labelers.

To address the above challenges, existing work on active learning with crowdsourcing labelers mainly follow two directions. One research direction utilizes relabeling strategy to obviate the effect of noise. Follow this idea, Sheng et al. [71] proposed a crowdsourcing resolution in supervised learning scenarios. Based on Sheng et al.'s work [71], Zhao et al. [72] applied crowdsourcing labelers in active learning framework by incremental relabeling only the most import instances. Fu et al. [79] proposed a new active learning paradigm, in which a nonexpert labeler is only asked whether a pair of instances belong to the same class. To instantiate the proposed paradigm, it adopts the MinCut algorithm as the base classifier and repeatedly updates the unlabeled edge weights on the max-flow paths in the graph. Finally, an unlabeled subset of nodes with the highest prediction confidence are added into labeled nodes.

Besides relabeling strategy, taking labeling cost into consideration is the other research direction. Integrating a cost budget into instance-selection metrics, it guarantees that the selected optimal subset subject to budget constraint, where budget is the total time cost available on annotation. Vijayanarasimhan et al. [81] formulates a budgeted selection task as a continuous optimization problem where the optimal selected subset maximizes the improvement to the classifier's objective, with a labeling cost budget constraint. Proactive learning [80] focuses on selecting an optimal oracle as well as an optimal instance at the same time using a decision theoretic approach.

6.4 Active learning for domain adaptations

It is desirable that a model built based on plenty labeled instances in one domain should perform reasonably well on data from different but similar domains [77]. For example, a classifier trained to classify "indoor" versus "outdoor" images should be beneficial for training classifier to classify "building" versus "natural scenery" images. A model on the source domain straightly applies in the target domain may result in a serious accuracy reduction. Therefore, we need to additionally label some instances in the target domain, so as to help leverage the original model apply in the target domain without performance compromise, which is known as domain adaptation. However, labeling a large amount of instances in the target domain is a costly and expensive process. So a promising resolution is seeking to minimize the amount of new annotation effort required to achieve good performance in the target domain. To reduce labeling cost, active learning can be employed to select instances to annotate form the target domain of interest.

Active learning in a domain adaptation setting has received little attention so far, where existing work mainly follows either pool-based or online active learning settings for domain adaption. In the pool-based active learning setting, Chan and Ng [73] proposes to combine active learning with domain adaptation for word sense disambiguation system. Shi et al. [75] employs an initial pool of labeled target domain to help train a in-domain model. In the online active learning setting, Saha et al. [74] presents a novel approach that use a domain-separator hypothesis in the active query process, and further leverage inter-domain information. Meanwhile, Zhu et al. [76] proposed a transfer active learning approach which actively selects (and labels) samples from auxiliary domains to improve the learning for a target domain. Both approaches in [74,76] can be used in pool-based or online active learning settings.

7 Conclusions

With the goal of labeling the most informative instances to achieve high prediction accuracies with minimum cost, active learning is a continuously growing area in machine learning research. In previous work, the emphasis has been on the design of new query strategies for instance-selection criteria. In this paper, we categorized existing query strategies in active learning into two groups: (1) Active learning based on IID instance uncertainty, and (2) Active learning based on instance correlations. We surveyed the two types of query strategies, analyzed and compared the time complexity of some representative methods, and briefly discussed some potential issues in the existing designs. A number of emerging active learning scenarios and new approaches are also discussed in this paper. Our survey, which mainly emphasizes on instance selection, provides a high-level summarization for interested readers to take instance correlations into consideration for designing effective active learning solutions.

Acknowledgments This research is sponsored by Australian Research Council Discovery Project funding scheme (Grant No. DP1093762) and by ARC Future Fellowship funding scheme (Grant No. FT100100971).

References

1. Aminian M (2005) Active learning with scarcely labeled instances via bias variance reduction. In: Proceedings of international conference on artificial intelligence and machine learning (ICAIML 2005), Cairo, pp 41–45
2. Becker M, Hachey B, Alex B, Grover C (2005) Optimising selective sampling for boostrapping named entity recognition. In : Workshop on learning with multiple view the 22nd international conference on machine learning (ICML 2005), Bonn, pp 5–11
3. Bilgic M, Mihalkova L, Getoor L (2010) Active learning for networked data. In: Proceedings of the 27th international conference on machine learning (ICML 2010), ACM, Haifa, pp 79–86
4. Bottou L (1991) One approche theorique del apprentissage connexionniste: applications. Ala reconnaissance de la parole. Doctoral dissertation, Universite de Paris XI
5. Burl MC, Wang E (2009) Active learning for directed exploration of complex systems. In: Proceedings of the 26th international conference on machine learning (ICML 2009), Montreal, pp 89–96
6. Campbell C, Cristianini N, Smola A (2000) Query learning with large margin classifiers. In: Proceedings of the 17th international conference of machine learning (ICML 2000), CA, pp 111–118
7. Carlson A, Berreridge J, Wang R, Hruschka ER, Mitchell TM (2010) Coupling semi-supervised learning of information extraction. In: Proceedings of the ACM international conference on web search and data mining (ICWSDM-2010), Washington, pp 101–110
8. Chang MW, Ratinov L, Rizzolo N, Roth D (2008) Learning and inference with constraints. In: Proceedings of the 23rd national conference on artificial intelligence (AAAI 2008), Chicago, pp 1513–1518
9. Chang MW, Ratinov LA, Roth D (2007) Guiding semi-supervision with constraint-driven learning. In: Proceedings of the 45th annual meeting of the association for computational linguistics (ACL 2007), Prague, pp 280–287
10. Chen Y, Subramani M (2010) Study of active learning in the challenge. In: Proceedings of the international joint conference on neural network (IJCNN 2010), Barcelona, pp 1–7
11. Cheng H, Zhang R, Peng Y, Mao J, Tan P (2008) Maximum margin active learning for sequence labeling with different length. In: Proceedings of the 8th industrial conference on advances in data mining: medical applications E-commerce marketing and theoretical aspects (ICADM 2008), Leipzig, pp 345–359
12. Copa L, Devis T, Michele V, Mikhail K (2010) Unbiased query-by-bagging active learning for VHR image classification. In: Proceedings of conference on image and signal processing for remote sensing XVI (ISPRS 2010), vol 7830, Toulouse, pp 78300K–78300K-8
13. Escudeiro N, Jorge A (2010) D-confidence: an active learning strategy which efficiently identifies small classes. In: Proceedings of workshop on active learning for natural language processing (ALNLP 2010), Los Angels, pp 18–26
14. Fine S, Bachrach RG, Shamir E (2002) Query by committee liner separation and random walks. Theor Comput Sci 284(1):25–51
15. Fuji A, Tokunaga T, Inui K, Tanaka H (1998) Selective sampling for example based word sense disambiguation. Comput Linguist 24(4):573–597
16. Gilad-Bachrach R, Navor A (2003) Kernel query by committee algorithm. Technology report no. 2003-88 Leibniz centre, The Hebrew University
17. Godec et al (2010) Context-driven clustering by multi-class classification in an active learning framework. In 2010 IEEE computer society conference on computer vision and pattern recognition workshops, pp 19–24

18. Hassanzadeh H, Keyvanpour M (2011) A variance based active learning approach for named entity recognition. In: Intelligent computing and information science, vol 135, Springer, Berlin, pp 347–352
19. Hoi SCH, Jin R, Lyu MR (2006) Large-scale text categorization by batch model active learning. In: The international conference on the world wide web (WWW 2006), ACM Press, New york, pp 633–642
20. Hoi SHC, Jin R, Zhu J, Lyu MR (2006) Batch mode active learning and its application to medical image classification. In: The 23rd international conference on machine learning (ICML 2006), Pittsburgh, pp 417–424
21. Holub A, Perona P (2008) Entropy-based active learning for object recognition. In: IEEE computer society conference on computer vision and pattern recognition workshop anchorage (CVPR 2008), pp 1–8
22. Huang A, Milne D, Frank E, Witten I (2008) Clustering documents with active learning using wikipedia. In: The 8th IEEE international conference on data mining (ICDM 2008), Pisa, pp 839–844
23. Huang J, Milne D, Frank E, Witten I (2007) Efficient multiclass boosting classification with active learning. In: The SIAM international conference on data mining (SDM 2007), Minnesota, pp 297–308
24. Ishihara T, Abe KI, Takeda H (1988) Extensions of innovations dual control. Int J Syst Sci 19:653–667
25. Jones R, Ghani R, Mitchell T, Rilo E (2003) Active learning for information extraction with multiple view feature sets. In: Proceedings of ECML Workshop on Adaptive Text Extraction and Mining (ATEM-2003)
26. Kim J, Song Y, Kim S, Cha J, Lee G (2006) MMr-based active machine learning for bionamed entity recognition. In: Human language technology and the North American association for computational linguistics, ACL Press, pp 69–72
27. Kunegis J, Lommatzsch A, Bauckhage C (2008) Alternative similarity functions for graph kernels. In: Proceedings of international conference on pattern recognition (ICPR 2008), Florida, pp 1–4
28. Lewis D, Gale W (1994) A sequential algorithm for training text classifiers. In: Proceedings of the ACM SIGIR conference on research and development in information retrieval (SIGIR 1994), Dublin, pp 3–12
29. Li B, Yu S, Lu Q (2003) An improved k-nearest neighbor algorithm for text categorization. In: Proceedings of the 20th international conference on computer processing of oriental languages (CPOL 2003), Shenyang, pp 12–19
30. Li D, Qian F, Fu P (2002) Variance minimization approach for a class of dual control problems. In: Proceedings of the 2002 American control conference (ACC 2002), Alaska, pp 3759–3764
31. Li M, Ishwar KS (2006) Confidence-based active learning. IEEE Trans Pattern Anal Mach Intell 28(8):1251–1261
32. Long B, Chapelle O, Zhang Y, Chang Y, Zheng Z, Tseng B (2010) Active learning for ranking through expected loss optimization. In: Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval (SIGIR 2010), Geneva, pp 267–274
33. Mann G, McCallum A (2007) Effiecent computation of entropy gradient for semi-supervised conditional random fields. In: Proceedings of the conference of the North American chapter of the association for computational linguistics (NAACL 2007), PA, pp 109–112
34. McCallum AK, Nigam K (1998) Employing EM in pool-based active learning for text classification. In: Proceedings of the international conference on machine learning (ICML 1998), Morgan, pp 359–367
35. Milito R, Padilla C, Padilla R, Cadorin D (1982) An innovations approach to dual control. IEEE Trans Autom Control 27(1):132–137
36. Muslea I (2002) Active learning with multiple views. Doctoral dissertation, University of South California
37. Nguyen HT, Smeulders A (2004) Active learning using pre-clustering. In: Proceedings of the 21st international conference on machine learning (ICML 2004), Banff, pp 839–846
38. Nguyen HV, Li B (2010) Cosine similarity metric learning for face verification. In: Proceedings of Asian conference on computer vision (ACCV 2010), QueensTown, pp 709–720
39. Olsson F (2009) A literature survey of active learning machine learning in the context of natural language procession. Swedish Institute of Computer Science, Technical report T2009:06
40. Qi G, Hua X, Rui Y, Tang J, Zhang H (2008) Two-dimensional active learning for image classification. In: Proceedings of the 23rd IEEE conference on computer vision and pattern recognition (CVPR 2008), Alaska, pp 1–8
41. Roy N, McCallum A (2001) Toward optimal active learning through sampling estimation of error reduction. In: Proceedings of the international conference on machine learning (ICML 2001), Morgan, pp 441–448
42. Saar-Tsechansky M, Provost F (2000) Variance-based active learning. The CeDER working paper no. IS-00-05
43. Seung H,S, Opper M, Sompolinsky H (1992) Query by committee. In: Proceedings of the 5th annual workshop on computational learning theory (COLT 1992), Pittsburgh, pp 287–294
44. Settles B (2010) Active learning literature survey. Technical report 1648, University of Wisconsin, Madison

45. Settles B, Craven M (2008) An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP-2008), Hawaii, pp 1070–1079
46. Settles B, Craven M, Ray S (2008) Multiple-instance active learning. Adv Neural Inf Process Syst 20:1289–1296
47. Shen D, Zhang J, Su J, Zhou G, Tan C (2004) Multi-criteria-based active learning for named entity recognition. In: Proceedings of the 42nd annual meeting of association for computational linguistics (ACL 2004), Barcelona, pp 589–596
48. Shi S, Liu Y, Huang Y, Zhu S, Liu Y (2008) Active learning for knn based on bagging features. In: Proceedings of the 4th international conference on natural computation (ICNC 2008), Jinan, pp 61–64
49. Shum S, Dehak N, Dehak R, Glass J (2010) Unsupervised speaker adaptation based on the consine similarity for text-independent speaker verification. In: Proceedings of the IEEE Odyssey workshop, Brno
50. Stolfo S, Fan W, Lee W, Prodromidis A (1997) Credit card fraud detection using meta-learning: issues and initial results. In: Proceedings of AAAI workshop on fraud detection and risk management (AAAI 1997), California, pp 83–90
51. Sun S (2010) Active learning with extremely sparse labeled examples. In: Proceedings of the 10th Brazilian symposium on neural networks (SBRN 2010), Sao Paulo, pp 2980–2984
52. Sohn S, Comeau D, Kim W, Wilbur W (2009) Term-centric active learning for naive bayes document classification. Open Inf Syst J 3:54–67
53. Wang M, Hua X (2011) Active learning in multimedia annotation and retrieval: a survey. ACM Trans Intell Syst Technolo 2(2):3–23
54. Wang Z, Song Y, Zhang C (2009) Efficient active learning with boosting. In: Proceedings of the SIAM data mining conference (SDM 2009), Nevada, pp 1232–1243
55. Weber JS, Pollack ME (2007) Entropy-driven online active learning for interactive calendar management. In: Proceedings of the 12th international conference on intelligent user interfaces (ICIUI 2007), Hawaii, pp 141–150
56. Wittenmark B (1975) An active suboptimal dual controller for systems with stochastic parameters. Automat Control Theory Appl 3:13–19
57. Zhu X, Ghahramani Z, John L (2003) Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of the 20th international conference on machine learning (ICML 2003), Washington, pp 912–919
58. Yan S (2005) Semi-automatic video semantic annotation based on active learning. Vis Commun Image Process 5960:251–258
59. Zhang Y (2010) Multi-task active learning with output constraints. In: Proceedings of the 24th AAAI conference on artificial intelligence (AAAI 2010), Georgia, pp 667–672
60. Zhao Y, Cao Y, Pan X (2008) A telecom clients credit risk rating model based on active learning. In: Proceedings of IEEE international conference on automation and logistics (ICAL 2008), Qingdao, pp 2590–2593
61. Zhao Y, Xu C, Cao Y (2006) Research on query-by-committee method of active learning and application. In: Lecture notes on artificial intelligence (LNAI 2006), vol 4093, pp 985–991
62. Zhou Z, Sun Y, Li Y (2009) Multi-instance learning by treating instances as non-i,i,d, samples. In: Proceedings of the 26th international conference on machine learning (ICML 2009), Montreal, pp 1249–1256
63. Zhu J, Wang H, Tsou B, Ma M (2010) Active learning with sampling by uncertainty and density for instances annotations. IEEE Trans Audio Speech Lang Process 18(6):1323–1331
64. Zhu X (2008) Semi-supervised learning literature survey. In: Computer sciences TR 1530, University of Wisconsin, Madison
65. Zhu X, Zhang P, Lin X, Shi Y (2007) Active learning from data streams. In: Proceedings of the 7th IEEE international conference on data mining (ICDM 2007), Nebraska, pp 757–762
66. Bilgic M, Getoor L (2010) Active inference for collective classification. In: Proceedings of the 24th AAAI conference on artificial intelligence (AAAI 2010), Georgia, pp 1652–1655
67. Chu W, Zinkevich M, Li L (2011) Unbiased online active learning in data streams. In: Proceedings of the 17th ACM SIGKDD conference on knowledge discovery and data mining (SIGKDD 2011), CA
68. Zhang P, Zhu X, Tan J, Guo L (2010) Classifier and cluster ensembles formining concept drifting data streams. In: Proceedings of the 10th IEEE international conference on data mining (ICDM 2010), Sydney, pp 1175–1180
69. Cesa-Bianchi N, Gentile C, Vitale F, Zappella G (2010) Active learing on trees and graphs. In: Proceedings of the 23rd international conference on learning theory, Haifa, pp 320–332
70. Guillory A, Bilmes J (2009) Labeled selection on graphs. In: Proceedings of 23rd annual conference on neural information processing systems (NIPS 2009), Vancouver, pp 320–332

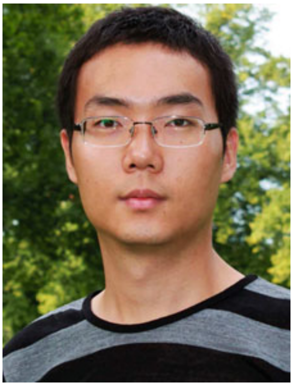
71. Sheng VS, Provost F, Ipeirotis P (2008) Get another label? Improving data quality and data mining using multiple, noisy labelers. In: Proceedings of 16th ACM SIGKDD conference on knowledge discovery and data mining (KDD 2008), Washington, pp 615–622
72. Zhao L, Sukthankar G, Sukthankar R (2011) Incremental relabeling for active learning with noisy crowdsourced annotations. In: Proceedings of the 2011 IEEE third international confernece on social computing (SocialCom 2011), Boston, pp 728–733
73. Chan Y, Ng H (2007) Domain adaptation with active learning for word sense disambiguation. Comput Linguist 45:49–56
74. Saha A, Rai P, Daume H, Venkatasubramanian S, DuVall S (2011) Active supervised domain adaptation. In: Proceedings of European conference on machine learning and principles and practice of knowledge discovery in databases (ECML/PKDD 2011), Athens
75. Shi X, Fan W, Ren J (2008) Actively transfer domain knowledge. In: Proceedings of European conference on machine learning and principles and practice of knowledge discovery in databases (ECML/PKDD 2011), Antwerp
76. Zhu Z, Zhu X, Ye Y, Guo Y, Xue X (2011) Transfer active learning. In: Proceedings of the 20th ACM international conference on information and knowledge management (CIKM 2011), Glasgow
77. Zhu X (2011) Cross-domain semi-supervised learning using feature formulation. IEEE Trans Syst Man Cybern B 41(6):1627–1638
78. Zhu X, Wu X (2006) Scalable representative instance selection and ranking. In: Proceedings of the 18th international conference on pattern recognition (ICPR 2006), Hongkong, pp 352–355
79. Fu Y, Li B, Zhu X, Zhang C (2011) Do they belong to the same class: active learning by querying pairwise label homogeneity. In: Proceedings of the 20th ACM conference on information and knowledge management (CIKM), Glasgow, pp 2161–2164
80. Donmez P, Carbonell J (2008) Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In: Proceedings of the ACM conference on information and knowledge management (CIKM 2008), pp 619–628
81. Vijayanarasimhan S, Jain P, Grauman K (2010) Far-sighted active learning on a budget for image and video recognition. In: Proceedings of the 23rd IEEE conference on computer vision and pattern recognition (CVPR 2010). San Francisco, pp 3035–3042
82. Abe N, Mamitsuka H (1998) Query learning strategies using boosting and bagging. In: Proceedings of the 15th international conference on machine learning (ICML 1998), pp 1–9
83. Bifet A, Holmes G, Pfahringer B, Kirkby R, Gavalda R (2009) New ensemble methods for evolving data streams. In: Proceedings of the 15th ACM SIGKDD conference on knowledge discovery and data mining (SIGKDD 2009), Paris, pp 139–148
84. Fan W, Huang Y, Wang H, Yu P(2004) Active mining of data streams. In: Proceedings of SIAM international conference on data mining (SDM 2004), Florida
85. Brecheisen S, Kriegel H, Pfeifle M (2006) Multi-step density-based clustering. Knowl Inf Syst 9(3): 284–308
86. Hovsepian K, Anselmo P, Mazumdar S (2011) Supervised inductive learning with Lotka–Volterra derived models. Knowl Inf Syst 26(2):195–223
87. Zhou Z, Li M (2010) Semi-supervised learning by disagreement. Knowl Inf Syst 24(3):415–439
88. Amini M, Gallinari P (2005) Semi-supervised learning with an imperfect supervisor. Knowl Inf Syst 13(1):1–42
89. Sinohara Y, Miura T (2003) Active feature selection based on a very limited number of entities. Adv Intell Data Anal 2811:611–622
90. Beygelzimer A, Dasgupa S, Langford J (2009) Important weighted active learning. In: Proceedings of the 26th international conference on machine learning (ICML 2009), Montreal, pp 49–56
91. Bishan Y, Sun J, Wang T, Chen Z (2009) Effective multi-label active learning for text classification. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (SIGKDD 2009), Paris, pp 917–925
92. Vijayakumar S, Sugyama M, Ogawa H (1998) Training instances selection for optimal generalization with noise variance reduction in neural network. In: Proceedings of the 10th Italian workshop on neural nets, Vietri sul Mare, Italy, pp 1530–1547
93. Culotta A, McCallum A (2005) Reducing labeling effort for stuctured prediction tasks. In: Proceedings of the 20th national conference on artificial intelligence (AAAI 2005), pp 746–751
94. Zhao W, He Q, Ma H, Shi Z (2012) Effective semi-supervised document clustering via active learning with instance-level constraints. Knowl Inf Syst 3(3):569–587
95. Zhu X, Ding W, Yu P, Zhang C (2011) One-class learning and concept summarization for data streams. Knowl Inf Syst 28(3):523–553

96. Pan S, Zhang Y, Li X (2011) Dynamic classifier ensemble for positive unlabeled text stream classification. Knowl Inf Syst 1–21. doi:10.1007/s10115-011-0469-2
97. Liu W, Wang T (2011) Online active multi-field learning for efficent email spam filtering. Knowl Inf Syst 1–20. doi:10.1007/s10115-011-0461-x

Author Biographies

Yifan Fu received her M.E. degree in Software Engineering from Northeast Normal University, Changchun China, in 2009. She is currently a Ph.D. student in the Centre for Quantum Computation and Intelligent Systems (QCIS), University of Technology, Sydney (UTS), Australia (since 2010). Her research interests lie in Machine Learning and Data Mining; more specifically, she is interested in active learning, ensemble mehods, and graph mining.

Xingquan Zhu received his Ph.D. degree in Computer Science from Fudan University, Shanghai China, in 2001. He is a recipient of the Australia ARC Future Fellowship and a Professor of the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS), Australia. Dr. Zhu's research mainly focuses on data mining, machine learning, and multimedia systems. Since 2000, he has published more than 120 referred journal and conference proceedings papers in these areas. Dr. Zhu is an Associate Editor of the IEEE Transactions on Knowledge and Data Engineering (2009), and a Program Committee Co-Chair for the 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2011) and the 9th International Conference on Machine Learning and Applications (ICMLA 2010).



Bin Li received his Ph.D. degree in Computer Science from Fudan University, Shanghai, China, in 2009. He is currently a Lecturer and previously a Postdoctoral Research Fellow in the Centre for Quantum Computation and Intelligent Systems (QCIS), University of Technology, Sydney (UTS), Australia (since 2011). Prior to this, he was a Postdoctoral Research Fellow at the Institut TELECOM SudParis, France (2009–2010). Dr. Bin Li's research interests include Machine Learning and Data Mining methods and their applications to social media mining, recommender systems, and ubiquitous computing.