Concept Sampling: Towards Systematic Selection in Large-Scale Mixed Concepts in Machine Learning

Yi Zhang$^{1, 2}$ and Xiaoming Jin^{1}
^{1}School of Software, Tsinghua University
^{2}Department of Computer Science, Tsinghua University
Beijing, 100084, China
zhang-yi@mails.tsinghua.edu.cn, xmjin@tsinghua.edu.cn

Abstract

This paper addresses the problem of concept sampling. In many real-world applications, a large collection of mixed concepts is available for decision making. However, the collection is often so large that it is difficult if not unrealistic to utilize those concepts directly, due to the domain-specific limitations of available space or time. This naturally yields the need for concept reduction. In this paper, we introduce the novel problem of concept sampling: to find the optimal subset of a large collection of mixed concepts in advance so that the performance of future decision making can be best preserved by selectively combining the concepts remained in the subset. The problem is formulized as an optimization process based on our derivation of a target function, which ties a clear connection between the composition of the concept subset and the expected error of future decision making upon the subset. Then, based on this target function, a sampling algorithm is developed and its effectiveness is discussed. Extensive empirical studies suggest that, the proposed concept sampling method well preserves the performance of decision making while dramatically reduces the number of concepts maintained and thus justify its usefulness in handling large-scale mixed concepts.

1 Introduction

In many real-world applications people in machine learning community are confronted with ***large*** numbers of ***mixed*** concepts[1], upon which the final decision is made. The inherent reason leading to this situation is that, in many cases, the concept underlying the data evolves due to the changes of hidden contexts [Widmer *et al*., 1996].

Example 1. Concepts related to stock trading strategies are influenced by many time-related factors which are hardly accessible, such as macro economic environments and political events [Harries *et al*., 1998]. On this occasion, one has to break training data into segments over short time intervals in order to extract stable concept from each interval, and then decision making is made upon the resulting large collection of diverse concepts. This strategy is widely used in on-line learning over data streams [Street *et al*., 2001; Wang *et al*. 2003] and result in unlimited numbers of mixed concepts.

Example 2. Consider the problem of detecting credit card frauds. The characteristics of the fraud events depend on some hidden contexts, such as the policies of the specific bank branch, the economic conditions, and the new law in the local area. Thus the concepts, i.e. the fraud patterns underlying the data, from different branches or even from different periods of the same branch may differ. Therefore, it is highly desirable to systematically analyze all these records. However, for privacy preserving, different branches can not share their records. In this case, one possible solution is that each branch periodically contributes the concept describing its recent records, and the models for decision making can be constructed upon the large collection of available concepts.

The typical way to utilize mixed concepts, when there are some data to be classified, is to select some "suitable" concept(s) and combine them by some strategies, which will involve searching within the large concept collection. However, such solutions are usually inefficient or even infeasible, because firstly, sometimes the collection is too large to be held in main memory, and secondly, the time complexity of decision making among large numbers of concepts is unacceptable for many efficiency-critical applications, such as online prediction of network intrusion.

This yields the need for concept reduction, the main focus of this paper. But is it possible to remove a large part of concepts in the collection while preserve the performance of decision making? The theory of ensemble learning (i.e. combining multiple concepts) [Kuncheva *et al*., 2003; Ruta *et al*., 2002] suggests that a concept can be approximated by an ensemble of similar concepts. Thus, the removed concepts can be estimated by selectively combining the remains.

While some literatures addressed the problem that seems similar to this topic, e.g. data sampling, to the best of our knowledge, concept sampling has not been explored. The main contributions of this paper are: (1) we propose the problem of concept sampling: to find the optimal subset of a

[1] In the field of machine learning, a *concept* corresponds to a learned description of a subset of instances defined over a large set [Mitchell, 1997]. More generally, concepts can be deemed as mappings from instance space to label space and represented by classifiers.

large collection of mixed concepts so that the performance of future decision making can be preserved. (2) We formally derive a target function that ties a clear connection between the composition of the subset and the expected error of the decision making upon this subset. Using this function, we formulize the problem of concept sampling as an optimization process. (3) We design a sampling procedure to determine the concept subset. The empirical results suggest that the proposed method well preserves the performance of decision making while dramatically reduces the number of concepts maintained, and thus justify its usefulness in handling large-scale mixed concepts.

2 Related Work

The existence of mixed concepts has long been accepted in machine learning, such as in the on-line learning problem over changing hidden contexts [Widmer *et al.*, 1996], or in the discussion of extracting stable concepts when hidden contexts evolve [Harries *et al.*, 1998]. Recently, on-line learning over data streams with evolving concept becomes an active field. Many algorithms in this field extract stable concepts from short time intervals and engage various forgetting policies to emphasize the recent concepts [Street *et al.*, 2001; Wang *et al.* 2003]. In fact, all this work is to deal with the mixed concepts. In this paper, we propose concept sampling, which plays an active role in managing large-scale mixed concepts. Even in data streams scenario, our method can act as an offline component to extract useful information from huge collection of historical concepts and thus complements the existing on-line learning styles.

Existing sampling techniques in machine learning mainly focus on instance space, which is to reduce the number of instances. General sampling methods often engage some empirical criteria on data distribution, e.g. density condensation [Mitra *et al.*, 2002], entropy-based data reduction [Huang *et al.*, 2006]. Similarly, specific methods, such as those for instance-based learning [Wilson *et al.*, 2000], also rely on some empirical criteria, e.g. assuming that object with the same label as its *k* neighbors is redundant.

Different from data sampling, this paper proposes a new problem of concept sampling, which is to preserve the quality of decision making upon the reduced concept set. More importantly, we derive a target function that ties a clear connection between the composition of the reduced set and the performance of decision making, and formulize the sampling as an optimization rather than relying on empirical criteria.

One of the foundations of our work is the theory of combining multiple concepts. Since different concepts exist, the diversity between concepts [Kuncheva *et al.*, 2003] must be considered when pursuing the consensus. Further, theoretical analysis about the performance of the ensemble classifiers [Ruta *et al.*, 2002] is engaged in our paper.

3 General Framework of Concept Sampling

Consider a large collection of mixed concepts $S = \{c_1, c_2, \ldots, c_n\}$, where each concept c_i is represented by a classifier, and Q denotes the unknown set of instances to be classified in the future. Essentially, S contains all the concepts so far observed, and we assume that for each future instance $q \in Q$, the correct concept c^q can be found in S[2].

However, S is often too large to be directly used due to the limitations of resources. Thus only a subset R whose size is much smaller is permitted. Since many concepts in S have to be removed, the right concepts c^q for many potential $q \in Q$ are not in R, and thus the performance of decision making declines. This calls for concept sampling, which is to reduce the number of concepts maintained but preserve the performance of decision making. This idea is possible because the theory of ensemble learning [Kuncheva *et al.*, 2003; Ruta *et al.*, 2002] suggests that a concept c can be approximated by an ensemble composed of its similar concepts.

The problem of ***concept sampling*** is formally defined as to find the optimal subset R with predefined size V_0 that satisfies

$$R = \arg\min_{R \subset S, |R| = V_0} (E(R,Q)) \tag{1}$$

where the target function $E(R,Q)$ is the expected error rates of using concept set R to classify instances in Q. Note that for many q in Q, the best fitted concepts c^q in the original set S have been removed, and we want to approximate these concepts by selectively combining the remained ones. Therefore, the target function $E(R, Q)$ can be further defined as:

$$E(R,Q) = \sum_{q \in Q} e(\varphi(R,q),q)p(q) \tag{2}$$

Here $p(q)$ is the probability of observing q in Q, $\varphi(R,q)$ is the set of concepts in R that are selected to classify q (i.e. the ensemble for classifying q), and $e(\varphi(R,q),q)$ is the expected error of this classification.

4 Concept Sampling Method

This section presents our concept sampling method. To minimize the target function mentioned in (2), the following key problems should be solved. First, the objective function (2) involves Q, the set of unlabeled instances in future, which is unseen in the time of sampling. Thus, the relationship between the composition of the subset R and the expected error defined in (2) is not clear. Second, sampling should be efficient to handle large concept collection. In section 4.1, we derivate an equivalent target function of (2) to deal with the first problem. In section 4.2, we design an efficient method to optimize this target function and discuss its effectiveness.

4.1 The Target Function

In this section, we derivate an equivalent target function for (2), which is computable given the subset R and the entire set S, and thus ties a clear connection between the composition of R and the expected error of future decision making upon R.

To handle the problem that Q is unknown, we define Q_c as the set of instances q in Q whose inherent concept c^q is c in S:

[2] In fact, handling the new concept never observed in S is investigated in the field of on-line learning with concept drift [Widmer *et al.*, 1996; Street *et al.*, 2001; Wang *et al.* 2003] and is beyond the scope of this paper.

$$Q_c = \{q \in Q \mid c^q = c\}, c \in S \tag{3}$$

Recall from section 3, the correct concept c^q of an unlabeled instance q in Q can be found in S. Thus, $\{Q_c \mid c \in S\}$ is a partition of Q. Then it holds that:

$$\begin{aligned} E(R,Q) &= \sum_{q \in Q} e(\varphi(R,q),q)p(q) \\ &= \sum_{c \in S} \sum_{q \in Q_c} e(\varphi(R,q),q)p(q) \end{aligned} \tag{4}$$

According to the definition of Q_c, the inherent concept c^q for each instance q in Q_c is the concept c in S. Thus, the ensemble of concepts that are selected by $\varphi(R,q)$ for classifying q should be the set of concepts that are chosen for approximating concept c. Moreover, the expected error of classifying q, termed $e(\varphi(R,q),q)$, can be represented by the expected error of the approximation on concept c. Therefore, given that $\pi(R,c)$ is the set of concepts in R that are selected to approximate concept c, and $\theta(\pi(R,c),c)$ refers to the expected error of this approximation, it holds that:

$$\begin{aligned} E(R,Q) &= \sum_{c \in S} \sum_{q \in Q_c} e(\varphi(R,q),q)p(q) \\ &= \sum_{c \in S} \sum_{q \in Q_c} \theta(\pi(R,c),c)p(q) \\ &= \sum_{c \in S} \theta(\pi(R,c),c) \sum_{q \in Q_c} p(q) \\ &= \sum_{c \in S} \theta(\pi(R,c),c)p(c) \\ &= E'(R,S) \end{aligned} \tag{5}$$

where three notations need investigation: $p(c)$, $\pi(R,c)$ and $\theta(\pi(R,c),c)$.

First, concepts in S are collected independently and assumed to be equally important. Thus, $p(c)$ is calculated as:

$$p(c) = \frac{1}{|S|} \tag{6}$$

Second, $\pi(R,c)$ defines the ensemble of concepts that can be used to approximate concept c. Since concept c can be approximated by combining multiple concepts similar to c given that these similar concepts make different mistakes [Kuncheva *et al.*, 2003; Ruta *et al.*, 2002], $\pi(R,c)$ is defined as the concepts in R that are, "by and large", similar to c:

$$\pi(R,c) = \{c' \in R \mid \theta(c',c) < d_0\} \tag{7}$$

Here threshold d_0 controls the strictness level of allowing a concept to be used as the ensemble member for approximating c. It is set as 0.15 in our empirical studies. Also note that $\theta(c',c)$, the expected error of using concept c' to approximate concept c, will be defined later. According to (7), concepts obviously inconsistent with c will be excluded, while concepts with slight deviation from c are permitted, in order to ensure the diversity in the ensemble.

Third, for $\theta(\pi(R,c),c)$, we start the discussion with the case that $\pi(R,c)$ returns a single concept c'. Thus, $\theta(c',c)$ is the estimated error of using c' to approximate c and can be obtained from empirical test. Given D, a set of instances that represents the instance distribution (labels are ignored), we use concept c to label the instances in D, and denote the labeled dataset as D^c. Then, $\theta(c',c)$ is naturally defined as:

$$\theta(c',c) = 1 - \rho(c', D^c) \tag{8}$$

where $\rho(c', D^c)$ measures the consistency (e.g. accuracy) of concept c' on D^c. Note that D can be obtained by random sampling on all the available instances, and can be incrementally maintained in dynamic situations [Vitter, 1985].

However, if $\pi(R,c)$ returns a group of concepts C^*, how to compute the expected error $\theta(C^*,c)$? In this paper we use the majority voting as the combination strategy since it is both theoretically sound and practically powerful [Kuncheva *et al.*, 2003; Ruta *et al.*, 2002]. It is true that $\theta(C^*,c)$ can be tested on D^c as in (8). But it is very time consuming: consider when searching the optimal R that minimizes (5), large numbers of possible combinations of $\pi(R,c)$ will be examined, and for each possible combination, we need compute $\theta(\pi(R,c),c)$. Fortunately, if the empirical error of each single concept c' on c has been calculated as (8), the exhaustive testing of $\theta(C^*,c)$ for all possible C^* can be replaced by theoretical estimation [Ruta *et al.*, 2002]: consider using an ensemble C^* of M concepts to approximate concept c. And e_i is the probability that the ith concept in C^* gives the incorrect label in each voting, which is obtained by (8). According to [Ruta *et al.*, 2002], the distribution of the normalized incorrect rates of these M concepts in each voting, defined as the number of incorrect votes divided by M, can be approximated by the probability density function of the normal distribution $f(x)$ whose mean and variance are:

$$\mu = \bar{e} = \frac{1}{M} \sum_{i=1}^{M} e_i, \nu = \frac{1}{M^2} \sum_{i=1}^{M} e_i(1-e_i) \tag{9}$$

Based on the above notions, $\theta(C^*,c)$, the expected error of the ensemble C^* via majority voting, is the probability that more than half of the M votes are incorrect:

$$\theta(C^*,c) = \int_{x>0.5} f(x)dx \tag{10}$$

Finally, by combining (5), (6), (7), (8) and (10), we get a computable target function which is equivalent to (2):

$$E(R,Q) = E'(R,S) = \frac{1}{|S|} \sum_{c \in S} \theta(\pi(R,c),c) \tag{11}$$

where $\pi(R,c)$ is defined as (7), $\theta(\pi(R,c),c)$ is computed as (8) when $\pi(R,c)$ returns a single concept and is estimated as (10) when $\pi(R,c)$ is an ensemble C^*.

Function (11) indicates that $E(R,Q)$ can be computed given the entire concept set S and a subset R. In this sense, it ties a clear connection between the composition of R and the performance of future decision making based on R. As a result, the sampling problem in (1) can be solved as an optimization process.

```
Input:  the concept collection $S$
        the desired size of the reduced set, denoted by $K$
        the empirical error between concepts as in (8)
        the threshold $d_0$ in (7)
Output: the subset $R$

Algorithm:
1. $R \leftarrow$ randomly sampling $K$ concepts from $S$
2. Compute $\theta(\pi(R,c),c)$ for each $c$ in $S$
3. Repeat until $R$ is invariant
   3.1 Find the concept $c'$ in $S$, that the reduction of (11) is
       largest if inserting $c'$ into $R$
   3.2 Insert $c'$ into $R$ and update $\theta(\pi(R,c),c)$ for each
       $c$ in $S$
   3.3 Find the concept $r$ in $R$ that the increase of (11) is
       smallest if removing $r$ from $R$
   3.4 Remove $r$ from $R$ and update $\theta(\pi(R,c),c)$ for
       each $c$ in $S$
   3.5 The removed $r$ is labeled so that it will not be in-
       serted into $R$ again.
```

Figure 1: Concept sampling algorithm

Intuitively speaking, which concepts should be in R? For each concept c in S, $\theta(C^*,c)$ computed as (9) and (10) reveals the relationship between the approximation error on c and the size M of the ensemble C^* to approximate c: if μ largely remains constant and is smaller than 0.5, the larger the M, the lower the ν, and thus the lower the approximation error in (10). In this sense, we want that each concept c in S has an ensemble C^* with enough qualified members from R.

On the one hand, the size of R is limited. Thus, the more ensembles that a concept r can join simultaneously, the more likely r should be in R. Note that whether a concept r can join the ensemble of a concept c is determined according to (7).

On the other hand, based on (9) and (10), the smaller the size M of the ensemble C^* of a concept c, the more dramatic the decrease of ν when inserting a new member to C^*, and thus the more substantial the decrease of the approximation error on c. Thus, the concept that can enter the ensembles that few other concepts can enter should be deemed valuable, and the concept that mainly joins the ensembles that many other concepts can also join is more or less trivial.

4.2 The Sampling Procedure

Clearly, it is very difficult to directly solve the optimization problem in (11) due to its nature of combinational optimization: there are almost infinite possible combinations of R given S. In this section, we proposed a feasible approach that iteratively improves the subset R, which is similarly in general to the methods designed for similar optimization problem, e.g., as in [Huang *et al.*, 2006]. More specifically, this method begins with a randomly selected subset R, and then successively improves it by firstly inserting into R an outside concept whose insertion maximizes the reduction of (11), and secondly, removing from R an inside concept whose deletion minimize the increase of (11). This process is repeated until the subset is invariant (e.g. invariant in 10 continuous iterations). The detailed algorithm is shown in figure 1.

According to the discussion at the end of section 4.1, the value of a concept r is determined by, firstly, whether it can join many ensembles for concepts in S, and secondly, whether there exist many other concepts in R that can also join these ensembles. In this sense, the step 3.1-3.4 of the algorithm repeatedly insert the concepts that join many ensembles that are in short of members, and remove the concepts in R that can enter few ensembles or that mainly enter the ensembles that already have sufficient members. Thus, the quality of R is continuously improved. Note that step 3.5 is used to avoid being trapped in the local minima.

Before formally analyzing the time complexity, we mention two points. Firstly, we assume that for each concept r, the notion $Nb(r) = \{c \in S \mid \theta(r,c) < d_0\}$ can be efficiently accessed: it can be computed before the sampling, and thus each time concept sampling is executed, $Nb(r)$ can be accessed directly. In fact, an equivalent notion of $Nb(r)$ is that $Nb(r) = \{c \in S \mid r \in \pi(R,c)\}$: the concepts whose ensembles admit r. Thus, when r is inserted into or removed from R, only concepts in $Nb(r)$ will have their ensemble changed. This will dramatically facilitate the evaluation of the change of (11) for possible insertions or deletions in R.

Secondly, the time-consuming integral computation in (10) depends on two variables, mean μ and variance ν of normal distribution $f(x)$. Thus we can discretize these two variables and then produce an integral table beforehand, from which (10) can be accessed directly.

Accordingly, time complexity of the proposed method is $O(nN)$, where n is number of iterations, and N is the size of S:

- Step 1: $O(N)$.
- Step 2: $O(N)$. Firstly, for each r in R, update μ and ν for concepts in $Nb(r)$; then for each c in S, compute $\theta(\pi(R,c),c)$ as (8) or (10); at last, compute (11). So the time complexity is $O(N)$.
- Step 3.1: $O(N)$. For each concept c' in S, inserting it to R only affects the concepts in $Nb(c')$, thus the reduction of (11) can be estimated quickly.
- Step 3.2: $O(|Nb(c')|)$.
- Step 3.3: $O(K)$. For each r in R, the increase of (11) is computed rapidly based upon $Nb(r)$. K is the size of R.
- Step 3.4: $O(|Nb(r)|)$.
- Step 3.5: $O(1)$.

5 Empirical Results

In this section, we present our empirical results. The goals of our experiments are: (1) to demonstrate the ability of our concept sampling method to preserve the performance of decision making while reducing the number of concepts maintained. (2) To justify the superiority of the proposed method over the straightforward selection method in terms of both performance and stability. We compared three methods: decision making upon the entire concept set S (*ES*), upon the reduced set obtained by concept sampling (*CS*); and upon the reduced set from random sampling (*RS*).

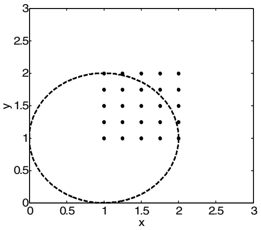


Figure 2: Distinct concepts scenario (scenario 1)

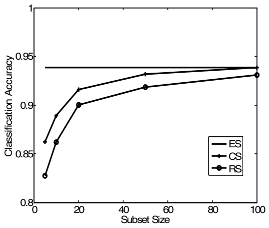


Figure 3: Accuracy in scenario 1

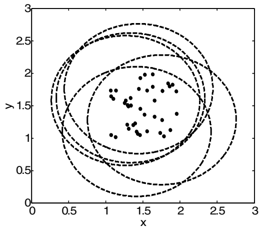


Figure 4: Miscellaneous concepts scenario (scenario 2)

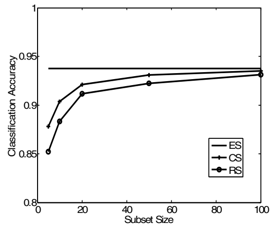


Figure 5: Accuracy in scenario 2

Given a concept set, the decision making strategy was: (1) for each instance q to be classified, a few evaluation data D^E (50 instances in our experiments) corresponding to the inherent concept c^q was given[3]; (2) based on D^E, the following "suitable concepts" were selected: the concept $c*$ that had the highest $\rho(c^*, D^c)$ (see (8)) plus all the concepts c satisfying $\rho(c, D^E) > 0.9\rho(c^*, D^E)$. (3) The selected concepts were combined by majority voting. In our experiments, each concept was represented by a C4.5 tree [Quinlan, 1993].

To comprehensively examine the performance of our concept sampling method in various applications, three typical scenarios were included in our empirical studies: (1) "distinct concepts scenario" and (2) "miscellaneous concepts scenario" are two boundary cases. Many real-world large collections of mixed concepts can be deemed as the "interpolations" of these two synthetic cases. Then, in (3) "real-world scenario", the real-world "Adult" dataset was tested, in order to evaluate the effectiveness of the proposed method in real-life applications.

Distinct Concepts Scenario: As in figure 2, 25 circle centers were produced. A concept was generated based on one of the centers: 2D points (x and y coordinates were both in [0, 3]) that fell into the circle around the center (with radius 1) were positive examples and otherwise negative. "Distinct concepts scenario" means that large numbers of concepts in the entire concept set S could be divided into distinct classes: concepts in the same class were similar, while concepts in different classes were distinct. Each of the 25 circle centers in figure 2 indicated a distinct class. For each class, 20 concepts were produced, each of which was trained from 200 random 2D points, and 5% noise on labels was added when training each concept. Thus, even concepts in the same class (i.e. determined by the same circle center) would be slightly different. Finally, 500 concepts in 25 distinct classes were generated, which formed the complete concept set S.

The entire set S, reduced sets R (generated by *CS*) and R' (generated by *RS*) were tested on 25 testing datasets, each containing 500 2D points corresponding to one distinct concept class. We focused on the average classification accuracies over these 25 datasets. For totally 500 concepts in S, diverse sampling rates were tested for both *CS* and *RS*, which resulted in concept subsets with 5, 10, 20, 50, 100 concepts, respectively. The final results were averaged over 10 independent runs.

Clearly, the ideal sampling algorithms should sensibly determine the number of remained concepts for each class. The results are shown in figure 3. It can be observed that: (1) *CS* method outperformed *RS*, on all the sampling rates, in term of the performance of decision making upon the reduced concept set. (2) The lower the sampling rate, the more obvious was the superiority of *CS* over *RS*. This is because when the number of concepts that can be retained is quite limited, the effectiveness of the sampling strategy becomes crucial: even the unsuitable allocation of one position will lead to remarkable performance degradation. (3) *CS* method with a sampling rate at 20% could provide largely the same classification accuracy as the original set S without sampling.

Miscellaneous Concepts Scenario: As in figure 4, 500 concepts were generated. But different from figure 2, no distinct class existed and the 500 concepts were miscellaneous, each of which corresponded to a random center circle. The entire set S, reduced set R and R' were tested on 500 testing datasets, each containing 500 examples corresponding to one concept. We compared the average classification accuracy over these 500 datasets. Also, different sampling rates were tested and the final results were averaged over 10 independent runs. From figure 5 we can observe the similar results as those in figure 3 (i.e. in distinct concepts scenario). And this justifies the superiorities of *CS* method over *RS* when concepts in the complete set S are miscellaneous.

Real-world Scenario**:** In this scenario, we tested the three methods on the real-world "Adult" dataset from UCI repository. We divided both the training and the testing datasets into eight groups, based on the value of "workclass" attribute. Two groups with very few examples were omitted. We extracted concepts from each of the remained six groups in the training dataset. Since each concept was trained from 100 examples in a group, groups with more examples produced more concepts. Totally 322 concepts were generated from the six groups, and formed the complete concept set S.

Testing dataset without the two omitted groups was directly used for test. This is reasonable because the proportion among the size of six remained groups is similar between the training dataset and the testing dataset. Thus, groups generating more concepts would have more instances in the testing dataset, which is consistent with the assumption that concepts in the complete set S should have the same probability to be useful "in future" (i.e. in testing dataset).

[3] In real-life applications, c^q is often estimated from evaluation data: in data stream scenario, the concept c^q of the current instance q is estimated from recent training examples. And in the credit card fraud scenario, the concept c^q of query q is estimated from recent labeled records from the branch where q is generated.

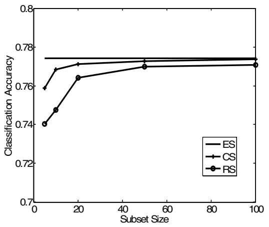
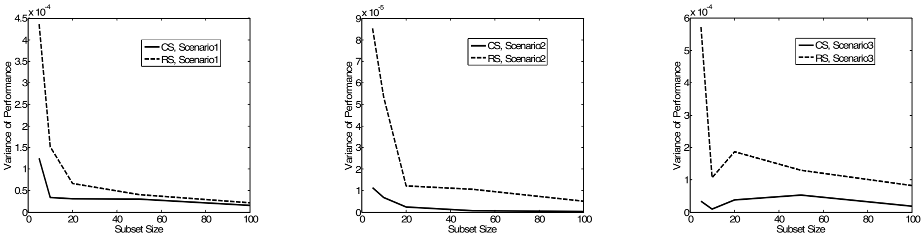


Figure 6: Accuracy on adult dataset (scenario 3)



(a) Scenario 1 **(b) Scenario 2** **(c) Scenario 3**

Figure 7: Variance of performance

Since label distribution in “Adult” dataset is unbalanced (24% positive examples), the “accuracy” was defined as: the average of the accuracy on the positive examples and the accuracy on the negative examples. Please note that this new measure should also be engaged as $\rho(c', D^c)$ in (8).

The final results averaged over 10 independent runs are shown in figure 6. Clearly, *CS* method obviously outperformed *RS* method, especially when the sampling rate is low. Secondly, using *CS* method, a subset of 20 concepts offered competent performance compared with the complete set of 322 concepts. This justified the usefulness of our approach in real world applications: well preserving the performance of the decision making while dramatically reducing the number of concepts maintained.

***Performance Variance*:** Stability is very important for sampling methods. While figure 3, 5 and 6 showed the average performance of *CS* and *RS* in 10 independent runs, figure 7(a), 7(b) and 7(c) focus on the variances of the performance in 10 runs. These results demonstrate that *CS* method is much more reliable than *RS* method. It is because the optimization of the proposed target function always guarantees the high quality of the reduced concept set.

6 Conclusion

In this paper, we introduced the novel problem of concept sampling: to retain an optimal subset from a large collection of mixed concepts to ensure efficient usage while guarantee that the performance of future decision making can be preserved by selectively combining the remained concepts. We provided a general framework of the problem, a target function that ties a clear connection between the composition of the concept subset and the error of future decision making, and an efficient sampling method based on the target function. The effectiveness and efficiency of the proposed method were discussed. Extensive empirical studies suggested that (1) the proposed method can well preserves the performance of decision making while dramatically reduce the number of concepts maintained; (2) it has superiorities over straightforward method in terms of both performance and stability. Through these studies, we demonstrated the usefulness of the proposed concept sampling method in handling large-scale mixed concepts.

Acknowledgments

The work was supported by Natural Science Foundation of China 60403021.

References

[Harries *et al*., 1998] S. B. Harries and C. Sammut. Extracting Hidden Context. *Machine Learning*, 32:101-126, 1998.

[Huang *et al*., 2006] D. Huang and T. W. S. Chow. Enhancing density-based data reduction using entropy. *Neural Computation*, 18: 470-495, 2006.

[Kuncheva *et al*., 2003] L. Kuncheva and C. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51: 181-207, 2003.

[Mitchell, 1997] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[Mitra *et al*., 2002] P. Mitra, C. A. Murthy and S. K. Pal. Density-based multiscale data condensation. *IEEE Trans. Pattern Analysis and Machine Intelligence.* 24(6): 734-747, 2002.

[Quinlan, 1993] J. R. Quinlan. *C4.5*: *Programs for Machine Learning*, Morgan Kaufman, 1993.

[Ruta *et al*., 2002] D. Ruta and B. Gabrys. A theoretical analysis of the limits of majority voting errors for multiple classifier systems. *Pattern Analysis and Applications*, 5: 333-350, 2002.

[Street *et al*., 2001] W. N. Street and Y. Kim. A streaming ensemble algorithm (SEA) for large-scale classification. In *Proc. of the 7th International Conference of Knowledge Disovery and Data Mining*, 377-392, San Francisco, CA, 2001.

[Vitter, 1985] J. S. Vitter. Random sampling with a reservoir. *ACM Trans. Mathematical Software*, 11(1), 37-57, 1985.

[Wang *et al*. 2003] H. Wang, W. Fan, P.S. Yu and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *Proc. of the 9th International Conference of Knowledge Discovery and Data Mining,* 226-235, Washington, DC, 2003.

[Widmer *et al*., 1996] G. Widmer and M. Kubat. Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning*, 23(1):69-101, 1996.

[Wilson *et al*., 2000] D. R. Wilson and T. R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38: 257-286, 2000.