Load-Balancing the Distance Computations in Record Linkage

Dimitrios Karapiperis  Vassilios S. Verykios
Hellenic Open University
School of Science and Technology
Patras, Greece
{dkarapiperis, verykios}@eap.gr

ABSTRACT
In this paper, we propose a novel method for distributing the distance computations of record pairs generated by a blocking mechanism to the reduce tasks of a Map/Reduce system. The proposed solutions in the literature analyze the blocks and then construct a profile, which contains the number of record pairs in each block. However, this deterministic process, including all its variants, might incur considerable overhead given massive data sets. In contrast, our method utilizes two Map/Reduce jobs where the first job formulates the record pairs while the second job distributes these pairs to the reduce tasks, which perform the distance computations, using repetitive allocation rounds. In each such round, we utilize all the available reduce tasks on a random basis by generating permutations of their indexes. A series of experiments demonstrate an almost-equal distribution of the distance computations, which can be considered as the fair distribution of computational resources such as space and time consumption.

1. INTRODUCTION
The process of merging disparate data sets in order to identify records that refer to the same real world entity is known as the record linkage, or the entity resolution, or the data matching problem. Basically, record linkage consists of two steps, namely the searching/blocking step and the matching step. In the first step, potentially matched pairs are formulated while in the second step these pairs are compared. For example, the standard blocking mechanism, which has been used extensively in the last decades, inserts all records that exhibit the same value in a chosen field(s) (attribute) into the same block. During the second step, the record pairs formulated are compared by using a metric that determines their distance. Thus, during the reduce phase, record pairs are formulated by the aggregated records in each block and their corresponding distances are computed. All tuples, aggregated by a blocking key, are forwarded to the tasks of the reduce phase where the tuples are aggregated by the same key. Then, summary operations are usually performed using the aggregated values of each key. As the sequence of the name Map/Reduce implies, the reduce phase is always performed after the map phase. Adjusting this model to the blocking step of record linkage, one can say that a key/value tuple consists of the blocking key as the key and the record as the value. Thus, during the reduce phase, record pairs are formulated by the aggregated records in each block and their corresponding distances are computed. All tuples, aggregated by a blocking key, are forwarded to a certain reduce task chosen (e.g., randomly or by hashing the blocking key) by the Map/Reduce system. This actually means that although each reduce task may have received record pairs corresponding to a certain number of blocks, being equal for all the reduce tasks, the actual load processed by each task, which is the volume of the distance computations, may exhibit large variations. As a result, some reduce tasks are overloaded by a large number of tuples and at the same time some others are underutilized due to the much smaller number of tuples forwarded. By assuming the standard blocking mechanism with the LastName field as the blocking key, values such as “Smith”, “Johnson”, or “Williams” will formulate overpopulated blocks while at the same time there will be blocks corresponding to less frequent values. This imbalance problem is known as the data skew issue of the reduce phase, which has as a consequence the poor utilization of computational resources such as space and time consumption.

In this paper, we propose a solution for the fair distribution of distance computations, which can be considered as the
load in the record linkage context, by using a Map/Reduce system. Our method utilizes two Map/Reduce jobs where the output of the first job is used as the input to the second job. During the first job, record pairs are formulated by applying a blocking mechanism while in the second job we distribute these pairs to the available reduce tasks for performing the distance computations. The distribution occurs by using these reduce tasks repetitively in allocation rounds. In each round, each record pair is assigned an index which represents a reduce task and is chosen randomly and uniformly. Then, each record pair is forwarded to the reduce task specified by that index so that the corresponding distance computation can be performed. This index is not used again until all indexes in the current round are exhausted. At that point, a new allocation round of record pairs to the reduce tasks begins. We provide both a theoretical analysis and a variety of experimental results that illustrate the efficiency of our scheme in minimizing the data skew in the Map/Reduce framework during the distance computations of record pairs formulated by a blocking mechanism.

The structure of the paper is organized as follows. Related work is described in Section 2. An outline of the processing tasks, which compose a Map/Reduce job, is given in Section 3 as well as we describe the data skew issue which is responsible for the imbalance problems in Map/Reduce. Our method is illustrated in Section 4 along with a theoretical analysis. Experimental results, including comparisons to other state-of-the-art methods, are presented in Section 5. Conclusions and ideas for future extensions are discussed in Section 6.

2. RELATED WORK

The Map/Reduce paradigm has been extensively used as a scalable tool for performing similarity joins in massive data sets. For example, Vernica et al. [13] present a framework of answering set similarity join queries in parallel by partitioning the data across nodes in order to balance the workload and minimize the need for replication. Blanas et al. in [1] demonstrate the performance of several join algorithms while in [9], Okcan and Riedewald implement arbitrary joins by using a single Map/Reduce job. They perform an analysis by sampling records in order to derive input statistics and minimize the job’s completion time. Especially in record linkage, the proposed solutions, which focus on the reduce phase data skew issue, employ a deterministic approach by running a separate Map/Reduce job that performs an analysis of block cardinalities on the considered data sets. This analysis is summarized in terms of a profiling data structure which could be materialized by a matrix utilized by the subsequent Map/Reduce jobs that perform the blocking and the matching step.

Authors in [7] calculate block cardinalities and store them in a matrix which is then shared by the subsequent Map/Reduce jobs. However, this method fails short during the matrix construction step given massive data sets. A scalable solution is presented in [14] where two low-memory consumption algorithms use block cardinality estimations provided by a sketch [4] in order to distribute the load accordingly. One of these algorithms is evaluated in Section 4.

SkewTune presented in [8] is an innovative solution for the reduce phase skew issue and works as follows. Whenever a node in the cluster is identified as idle, the task with the greatest expected remaining processing time is chosen and its unprocessed input data is redistributed so that all nodes are fully utilized. This redistribution though may incur considerable latency especially in cases where large chunks of data should be relocated.

In [6], authors propose a strategy for de-duplicating a redundant blocking mechanism, based on the Locality-Sensitive Hashing technique [5], by utilizing two Map/Reduce jobs. The first job formulates the record pairs while the second one eliminates the duplicate pairs.

3. PROBLEM FORMULATION

For illustration purposes and without loss of generality, we assume that two data custodians, namely Alice and Bob, are about to merge their data sets, $A$ and $B$ respectively, in order to find similar records which might represent common entities. We also assume a trusted third party, namely Charlie, who offers linkage services by utilizing a robust computational infrastructure which is capable of handling massive data sets. The data custodians submit to Charlie these data sets, which follow a common schema, and assign to him the task of identifying the matched record pairs.

Charlie, in order to avoid comparing each record of $A$ with each record of $B$ ($A \times B$), generates record pairs by applying the standard blocking mechanism as follows. A blocking key for each record is specified as the value of a chosen field, or fields, of the common schema. All records that exhibit the same blocking key value are inserted into the same block. The record pairs formulated within each block are compared in order to determine their distance. However, when we deal with massive data sets these record pairs could be in the order of millions or billions. For this reason, Charlie maintains a cluster of commodity hardware with a distributed file system that runs on top of it in order to leverage scalability of the required computations. The data sets are segmented by the distributed file system into equal subsets which are then distributed to the compute nodes of the cluster.

During processing, specialized tasks run on each node that perform computations by using only the local subset of records on each node. By assuming $N$ compute nodes, the data sets $A$ and $B$ are segmented into $N$ subsets which are distributed to the compute nodes. We have to note, that each such subset consists of records of both data sets $A$ and $B$. On each node, a task is initiated, that is $M_i$, which uses for each record the value of the blocking key to form a tuple $<K,V>$ where $V$ could be the record itself or an Id that references this record $i$. Then, on each node a task denoted by $P_i$ receives as input the output of the local $M_i$, which is a volume of $<K,V>$ tuples, and for each tuple it generates an integer value according to the key $K$. This integer value represents the task of the following phase to which this tuple should be forwarded. Therefore, pairs exhibiting the same key $K$ will be forwarded to the same task. Finally, this latter family of tasks receive the forwarded pairs, aggregate those pairs which exhibit the same $K$, and then perform the distance computations. All these tasks run in three distinct phases and they are coordinated by the Map/Reduce system

\[\text{http://hbase.apache.org/}\]
This workflow constitutes a Map/Reduce job, denoted by J. Complex computational problems though cannot be solved by a single job instead should be broken down into simpler subproblems where each subproblem should correspond to an individual job. Thus, a pipeline of chain jobs is executed where the output of a job is used as the input to the next job in line.

Tasks in each phase run independently and by no means do they communicate with one another. Although, this independence makes the whole system quite robust, because the failure of a task does not affect the execution of the other tasks, the distribution of load by the P's to the R's becomes a non-trivial issue. By taking into account that (a) the number of the formulated record pairs in certain blocks may be excessively large and (b) the independence of each R, we will end up with some R's, which will receive the load of the overpopulated blocks and will dominate the matching step, and a number of underutilized R's with much fewer records forwarded. This imbalance is commonly known as the data skew issue of the reduce phase and results in severe performance degradation of the record linkage process.

4. A PERMUTATION-BASED LOAD-BALANCING METHOD

Our Permutation-Based Load-Balancing method (PBLB) utilizes two Map/Reduce jobs, namely J_1 and J_2. Job J_1 formulates the Id pairs as described in Section [3] while the second job J_2, which is executed after J_1 has been completed, distributes almost equally these pairs to the R's, performs the corresponding distance computations. The M's of J_1 create a <K_1, V_1> tuple from each record, where K_1 is the blocking key value and V_1 is the value of the Id field of that record. All tuples associated with a certain value of K_1 are forwarded to a single R_j by using a simple hash mechanism which will be explained in Section [4]. Then, each R_j formulates Id pairs by using those tuples which belong to different data sets. The output of each R_j of J_1, which is a volume of Id pairs each denoted by W_i (w_i = |W_i|), is used as the input to the M's of J_2. Each such M, for each pair of W_i formulates a <K_2, V_2> tuple where K_2 holds the pair of Id values and V_2 represents the index of the R_j to which this pair will be forwarded. The choice for the value of V_2 occurs on a random basis in repetitive allocation rounds as follows. For each pair, we pick randomly, uniformly, and without replacement an index from the interval I = [0, N_R]. The chosen index represents an available R_j and it is assigned to V_2. When all indexes are exhausted, which implies that all available R's have been chosen, a new allocation round begins. The corresponding P's simply forward each Id pair to the R_j specified by the value of V_2. Finally, the R's of J_2 perform the distance computations by using the Ids held by K_2 in order to retrieve the corresponding records from the data store.

In each round, by choosing randomly and uniformly the index for each R_j from I without replacement, we actually generate permutations using the indexes of the R_j's included in I. These permuted indexes guarantee a fair distribution of load for the first q_p pairs where q_p = |W_i|/N_R. The remainder load of each W_i, denoted by r_W_i, and consists of n_i = w_i mod N_R pairs, is also distributed evenly to the available reduce tasks with high probability as shown theoretically in Section [4] and experimentally in Section [5]. These rounds are illustrated in Figure [1] where the 3rd round handles the remainder load. Figure [2] outlines the execution of jobs J_1 and J_2 including all the intermediate phases described before.

Algorithm [4] is executed by each M_i of J_2. An allocation round is implemented by using variables c and S. Both are used to store information about the R_j's chosen in an allocation round. More specifically, c is a counter, in the context of each M_i, that stores the number of the R_j's chosen thus far and S represents the set of the indexes of those R_j's.
randomly chosen index from $I$ is assigned to $V_2$ provided that this index has not been used before in the current allocation round. As shown in line 10 when $c$ gets equal to the total number of the reduce tasks, a new round of assignments begins by initializing $c$ to 0 (line 11) and $S$ to the empty set (line 12). By doing so, we generate permutations of the indexes of the $R_i$s giving an equal probability to an $R_j$ to be chosen.

4.1 Theoretical analysis

By using the repetitive allocation rounds, described in the previous subsection, we achieve an absolutely equal distribution for the first $q_i$ pairs of each $M_i$ to the $R_i$s. The number of the $R_i$s which will receive the remaining $n_i$ pairs depends on the value of $n_i$, where $n_i < N_R$.

**Theorem 1.** The probability for an $R_j$ to be chosen by an $M_i$ when distributing the remainder load $rW_i$ is:

$$\Pr[C_{M_i}(R_j)] = \frac{n_i}{N_R},$$

where $C_{M_i}(R_j)$ denotes the corresponding event.

**Proof.** During the distribution of each remainder load $rW_i$, the probability that a certain $R_j$ is not chosen by an $M_i$ in the first choice is $1 - (1/N_R)$, by using the complement of $C_{M_i}(R_j)$. In the second choice, it becomes $1 - [1/(N_R - 1)]$ since we have already used an $R_j$ which is not replaced. Generally, the probability of $C_{M_i}(R_j)$ given $n_i$ pairs is:

$$\Pr[C_{M_i}(R_j)] = 1 - \left(1 - \frac{1}{N_R}\right)(1 - \frac{1}{N_R - 1})\ldots(1 - \frac{1}{N_R - n_i + 1}) = \frac{N_R - n_i}{N_R} = \frac{n_i}{N_R}.$$
5. EVALUATION

We evaluate our method by applying the standard blocking mechanism using data sets extracted from the NCVR list. Each record of these data sets includes 5 fields, namely the Id, the LastName as the blocking key, the FirstName, the Address, and the Town. The imbalance ratio, which is equal to \( \max_{j=0}^{N_R-1} l_j / \sum_{j=0}^{N_R-1} l_j / N_R \), where \( l_j \) denotes the number of \( Id \) pairs forwarded to an \( R_j \), is used to measure the efficiency of each method at reducing the load imbalance among the utilized reduce tasks. The optimal value for the imbalance ratio is 1 because if the maximum load distributed to a reduce task is equal to the average load, then the optimal distribution has been achieved.

We compare our method with the naive random-selection scheme, termed as RND, and with a method introduced in [14], termed as SKETCH, where profiling information of block cardinalities is used. For method RND, we employ both jobs \( J_1 \) and \( J_2 \) but during the execution of \( J_2 \), the selection of the reduce task for each \( Id \) pair is made randomly and uniformly from \( I \) with replacement. Therefore, each reduce task is chosen with probability \( 1/N_R \). The power of this method is its simplicity in terms of implementation because its only requirement is a simple random function returning integer values from \( I \). Integral components of method SKETCH are specialized data structures, commonly known as sketches, built by fast and small-space algorithms for summarizing massive data in order to provide estimates for cardinality queries. More specifically, the Fast-AGMS sketches [4] are used in order to build profiles of estimated block cardinalities for record pairs which reside on each node of the cluster. These sketches are sent to a single reduce task that combines them and produces a vector where each cell may hold the estimated load for multiple blocks. We evaluate the Cell Block Division algorithm where a division process is applied to those cells which hold large loads in order to divide them into sub-cells before assigning them to the reduce tasks. We set both accuracy parameters \( \epsilon \) and \( \delta \) to 0.01. This sketch-based implementation is scalable to massive data sets mainly because it consumes a small amount of memory in order to store the profiling information.

![Figure 3: The frequency of the surnames in our data sets indicates a skewed distribution.](http://www.app.sboe.state.nc.us/data)

Apache Hadoop version 1.0.4 is used as the Map/Reduce implementation running on a cluster of eight compute nodes which are virtual machines of the cloud service of the Greek Research and Academic Community (Okeanos). The software components are developed using the Java programming language version 1.7.

We evaluate the above-mentioned methods by increasing the number of reduce tasks and the size of the data sets at hand performing two series of experiments. In the first series, we compose data sets by extracting records from the NCVR list without inserting any synthetic records. Figure 3 indicates a skewed distribution of surnames where the relative frequency of the first-ranked surname is equal to 0.013 while the frequency of the 200th-ranked surname is equal to \( 8.6 \times 10^{-4} \).

![Figure 4: Evaluating the performance of each method by using data sets extracted from the NCVR list.](http://hadoop.apache.org/)

(a) By increasing the number of reduce tasks PBLB exhibits a slight deviation from the optimal value (900K records for each data set).

(b) The performance of PBLB is near-optimal regardless of the size of the data sets used (32 reduce tasks).

![Figure 5: Evaluating the performance of each method by using synthetic blocks where their cardinality follows the Zipf distribution.](https://okeanos.grnet.gr/home/)

(a) Our method is not affected by the skew applied (32 reduce tasks - 500K records for each data set).

(b) Performance of SKETCH improves as skew increases (64 reduce tasks - 1000K records for each data set).

Figure 4 clearly shows that our method does not cause large variations in the distribution of load, exhibiting results very close to the optimal value. Method RND exhibits better results as the number of reduce tasks increases. This happens because the probability of skewing the load, which essentially is the number of collisions exhibited among the chosen reduce tasks, minimizes as the number of reduce tasks increases. Method SKETCH relies on block cardinality estimates which may deviate from the exact values especially...
in cases where the skew is not high enough, as reported in [12]. These deviations are reflected to the distribution of load affecting the performance negatively. In Figure 10, we illustrate experimental results by using 32 reduce tasks and increasing the size of the data sets that participate in the linkage process. The performance of PBLB is very close to the optimal value regardless of the size used. Also, method RND seems to converge to the optimal value (it converges asymptotically) as the size increases.

For the second series of experiments, we generate synthetic blocks by adding record pairs to those blocks generated by the first series of experiments so that their cardinality follows the Zipf distribution. More specifically, the cardinality of the \( k \)-th most populated block with record pairs is proportional to \( k^{-z} \) where \( z \) is the skew parameter. The performance exhibited by our method is not affected by the skew applied, as shown in both Figures 10a and 10b. On the contrary, the performance of method SKETCH improves as the skew increases. This implies the increased accuracy of estimates for block cardinalities provided by the sketches. Method RND is not affected by the skew applied but its performance is worse than the other methods. In Figures 6 and 7, we measure the execution time for all methods by increasing the number of reduce tasks. For these experiments, we used up to 16 reduce tasks because we had 8 virtual machines available and we configured Apache Hadoop to run at most 2 reduce tasks on each machine. Our method exhibits better results as the number of reduce tasks increases without being affected by the skew applied. Increased skew \( (z = 1.5) \) seems to affect positively the execution time of method SKETCH, as shown in Figure 7 due to more accurate profiling information which results in better utilization of the reduce tasks.

We also ran experiments using the default hash-based \( P_s \) used by Apache Hadoop. The map tasks block the records while the reduce tasks perform the distance computations of the record pairs formulated in each block. The default \( P_s \) in Apache Hadoop, for each tuple \( <K,V> \) map \( K \) the index of the reduce task to which this tuple will be forwarded. A hash function of the form \( h(K) \mod N_R \) is used where \( h(\cdot) \) produces an integer value by \( K \). The imbalance ratio using the data sets from the first series of experiments is constantly above 1.6 and becomes even higher, namely above 1.8, by using the synthetic data sets with skew parameter \( z = 1 \).

This imbalance is illustrated in Figure 8a as contrasted to the uniform load distribution achieved by PBLB, which is shown in Figure 8b.

6. CONCLUSIONS

In this paper, we introduce a simple, yet efficient, method for distributing the load of distance computations of record pairs generated by a blocking mechanism on an equal basis among the available reduce tasks by using the Map/Reduce framework. Our method does not require any profiling information but relies on a simple randomization scheme which applies a permutation-based mechanism in order to assign a distance computation to an available reduce task. Further research directions include the mining of statistics regarding characteristics of the considered data sets, as authors do in [9], so that by utilizing a single job, instead of two, we would both formulate and distribute the record pairs to the reduce tasks saving valuable amount of running time.

7. REFERENCES


ABSTRACT
Community Question Answering websites (CQA) offer a new opportunity for users to provide, search and share knowledge. Although the idea of receiving a direct, targeted response to a question sounds very attractive, the quality of the question itself can have an important effect on the likelihood of getting useful answers. High quality questions improve the CQA experience and therefore it is essential for CQA forums to better understand what characterizes questions that are more appealing for the forum community. In this survey, we review existing research on question quality in CQA websites. We discuss the possible measures of question quality and the question features that have been shown to influence question quality.

Keywords
Question quality, Community Question Answering, question features

1. INTRODUCTION
The web has changed the way people provide, search and share information and knowledge. It has become straightforward to submit keywords in a search engine to express a need, and the search engine immediately lists a large number of more or less relevant webpages from which the user can choose. However, the search results may not provide an exact solution to the user’s problem and it may be time-consuming to review all of them, without having a guarantee of finding the desired answer. Community Question Answering websites offer a new opportunity to obtain the desired knowledge in a more rapid and efficient way.

Community Question Answering (CQA) websites provide an interface for users to exchange and share knowledge. The users in such a forum can be divided in three groups: 1) users who only ask questions, 2) users who only answer questions and 3) users who ask and answer questions [1]. The user asking a question lacks knowledge of a specific topic and searches for an expert on the same topic to provide the desired knowledge. In this way, an asker is querying a topic and the experts providing the knowledge about this topic are the source of information, thus replacing other sources like documents or databases. Although the idea of receiving a direct response to a certain information need sounds very appealing, CQA websites also involve risk because the quality of the provided information is not guaranteed. An important difference between user-generated content and traditional content is the range of the content quality: user-generated content shows a higher variance in quality than traditional content [2], [4]. The quality distribution varies from very high to very low.

As there is a large number of CQA websites, it is important for a CQA website to provide high-quality content to distinguish itself from other websites. The importance of high-quality content in community-driven question and answering websites has been recognized and investigated in several studies. Importantly, in [2] is has been shown that there is a correlation between the question quality and answer quality- good answers are more likely to be given in response to good questions. Similarly, bad answers appeared in response to bad questions. According to the definition in [14], high quality questions are expected to draw greater user attention, to have more answer attempts and to obtain the best answer within a short period of time. High-quality questions thus help to improve the CQA website’s popularity as they, on the one side, contribute to efficient problem solving, and on the other side, enrich the community knowledge.

With the increase in popularity of CQA websites, not only the number of questions and the number of new members increased, but also the number of unanswered questions became high. For example, in Stack Overflow, a programming CQA forum, approximately 45 questions per month remained unanswered, according to statistics from 2012 [6]. By March 20, 2014, the number of unanswered questions was 752,533 out of 6,912,743 (approximately 10.9%). In Yahoo! Answers also approximately 15% of the incoming English questions remain unanswered [22]. Interestingly, the fact that those questions are not answered is not caused by users not having seen them. In fact, unanswered questions are seen 139 times on average [6]. A number of studies have attempted to predict whether a certain question will receive an answer or not and to determine the features that define question quality. In this survey, we will discuss the different measures used to evaluate question quality in CQA websites and will present the features of questions used in previous work that influence the question quality.
2. PROBLEM DESCRIPTION

The ability to recognize high quality questions is of high importance for both asker and answerer in a CQA website. In a good forum there should be a supply and demand balance – the aim of an asker is to receive a good answer to her question and the aim of a person providing an answer is to give a satisfying answer in order to increase her reputation. Therefore, it is essential for CQA forums to better understand what characterizes questions that are more appealing to be answered. A well-formulated question will increase the answerer’s willingness to answer and will help her to give an appropriate answer which, at the same time, will increase the asker’s satisfaction. It can also increase the question score and, respectively, the user’s reputation which would motivate her to increase her participation and privileges on the programming CQA website Stack Overflow. Seen from the asker’s side, not receiving an answer, or only a limited number of answers, is not only disappointing, but may also be of educational or professional disadvantage. What is more, even if a question finally receives an answer, the longer it takes to get an answer, the more likely that the answer quality will not be satisfactory.

Question quality is important not only for the personal use but also for question and answering platforms as a whole because appropriate question-answer pairs attract users and improve platform traffic. The aim of the asker is to obtain the information she is searching for and the aim of the person providing an answer is to give a satisfying answer to increase her reputation. For example, in StackOverflow there are a lot of questions with differing numbers of answers with a large variation, and it is observed that low-quality questions receive low-quality answers. Low-quality questions also affect the user’s experience on question and answering website. Furthermore, high-quality questions improve the entire question and answering platform as these questions are more appealing to users to share their knowledge and in this way to improve the overall platform knowledge. Finally, high-quality questions improve question retrieval and question recommendation in question and answering websites.

Several studies have focused on the quality of answers on CQA websites. More recently, there has been increasing interest in the quality of questions. A variety of metrics of question quality has been used, and accordingly, conclusions about which features influence question quality also differ. For example, in the authors find that short and long questions are more likely to be answered, whereas other studies state that too short questions have a low probability of obtaining an answer. Therefore, the aim of this survey is to provide an overview of the different measures to predict question quality, and the different features used in previous work that determine question quality.

3. QUESTION QUALITY

Different studies employ different definitions of question quality. In this section we describe the measures most commonly used in research on define question quality - the number of answers and the question score. Finally, additional measures for question quality used in some research will be summarized.

The number of answers is one of the clearest reflections of how interesting and useful a posted question is to the CQA community. In a domain-specific QA website, users share expertise knowledge, mostly in the form of question-answer pairs. These question-answer pairs are saved on the website and usually ranked by search engines which makes them retrievable and valuable for future information needs. A question has long term value when it draws the users’ attention long into the future after it was posted. Research has shown that the number of answers is the most significant feature to predict the long term value of a question together with its answers set. The number of answer is direct feedback on the usefulness/quality of the question. If the users assess it as being off topic or for some other reason inappropriate for the question answering community, they will be less likely to provide an answer.

Other research focuses not on the number of answers but on whether a question received at least one answer. Similarly, in a qualitative study is conducted on unanswered questions. Although one might think that questions remain unanswered because they were for some reason not discovered, the authors found that unanswered questions were on average seen 139 times. A second measure of the community response, indicating whether a question has been valuable for the question and answering community, is the question score. The question score consists of the sum of the upvote count and the (negative) downvote count. In forums like Stack Overflow, Quandora and Askbot, a question can be rewarded by voting it up. Equivalently, if the question was of low quality or useless to the community, the asker can be punished by having it voted down or even deleting it from the website. In general, answered questions on Stack Overflow have higher scores compared to unanswered questions. It has been shown that prior up and down votes correlate with the asker’s current question score. Hence, the question score is an appropriate feature to measure how the community users assess the question quality.

As both the question score and the number of answers are considered quality determinants, one would expect that a question with a high score receives many answers. That would be the case when a question is found very interesting and valuable to the community and if there were enough experts to answer it. Also if a question was not appropriate for the CQA community, it may not receive an answer and get a lot downvotes. However, the question score and the number of answers may not necessarily correlate. A question may address a new development or topic that is very interesting to the community but at the same time also very difficult to answer as there may be not enough experts familiar with it. Such a question may receive no answers but a lot of upvotes. If however a question was too easy or posted previously it may receive answers, but may not be evaluated highly as it does not contribute to the question answering website.

A number of other measures of question quality have been used in the literature. In a survey is conducted among professional software developers to identify code-related questions they find difficult to answer. In the focus is also on the content quality of Stack Overflow, but instead of investigating the features characterizing well-formulated questions, they concentrate on the features that describe low-quality questions. In Stack Overflow, if questions are off topic or of poor quality, they can be deleted by Stack Overflow moderators, experienced users with high reputation, or by the user who posted the question. As a deletion of a question is direct feedback regarding its quality, the aim in [10] is to find out what defines a question that is considered bad

1 http://stackoverflow.com/
2 http://www.quandora.com/
3 https://askbot.com/
4. FEATURES DETERMINING QUESTION QUALITY

The features which have influence on the question quality can be divided in two groups: question-related and asker-related attributes. The group of the question-related features is represented by the features tags, terms, question title and question body length and the presence of an example, in the case of Stack Overflow - a code snippet. Regarding asker-related features, the reputation of the user can be taken into consideration. Since a user would better understand the question quality at the moment of posting, we focus on features that relate to information that is available at the moment a question is posted, i.e. those features that contain information that only becomes available once it is already known whether and how many answers a question received, are left out. The reasoning behind this choice is that features which are not available at the moment of the posting cannot help the asker to improve his or her question [9, 10].

4.1.1 Question Related Features

4.1.1.1 Tags and Terms

In many QA forums, the asker can add tags to her question to indicate to which topic(s) the question is related. Intuitively, one would expect that some question topics will elicit more answers than others, just because more people might be working on a certain topic, i.e. there will be more potential answerers available. Although tags may potentially differentiate between the number of answers, the large number of unanswered questions cannot be explained by a lack of sufficient experts for certain topics [21]. The assigned tags are considered as representative topics and investigated tags used for unanswered questions but not for answered questions. The authors of this study found 274 unanswered topics linked to only 378 questions in total. The number of questions with these specific tags is very small compared to the total number of unanswered questions which would indicate that there is at least one expert for each tag/topic. However, as users mostly assign several tags to a question, covering very general to specific tags, the large number of unanswered questions cannot be explained by a lack of experts [21]. In [10] tags are also analyzed to investigate the topics of questions in Stack Overflow. They found that approximately ten percent of the tags found in deleted questions were not present in closed or regular questions. These questions, tagged for example as homework, job-hunting and polls, are beyond the interests of the programmer community. In both, [21] and [10], the authors assume that tags are representative of the actual question topics. According to [6], however, incorrect tagging is one of the characteristics of unanswered questions. In [18] an investigation is performed of the, among others, relationship between characteristics and the question type in Stack Overflow data. The question types are described based on two dimensions – the question topic and the main concern of the asker. The former is described by the technology or construct the user is asking about, and the latter dimension concerns the problem the asker wants to solve. In [18] the following question types are considered based on the problem of the asker: debug/corrective, need-to-know, how-to-do-it, seeking different-solution. The authors found that the answer attributes are likely to be determined only by the second question dimension, the main concern of the asker. In [10] the authors observed that a high percentage of author-deleted questions are marked as too localized and off topic, and that a high percentage of moderator-deleted questions are marked as subjective and not a real question. These results indicate that question topics, i.e. tags, may either be incorrect and/or may not be fully informative of the likelihood of receiving an answer, the number of answers, or question score.

A number of recent studies tried to infer question topics from the natural language used to formulate the questions. In [25] the contents of thousands of questions and answers on Stack Overflow are analyzed. The authors assume the number of the topics to be equal to five and use Latent Dirichlet Allocation (LDA) [78] to find latent topics. They manually label each of these topics to: user interface, stack trace, large code snippet, web document, miscellaneous. Table 1 shows the topics with the representative key words. The results if the study showed that the category of miscellaneous topics which consists of many different kinds of questions, holds the largest number of questions. The second largest category is the web document topic, followed by large code snippet, stack trace and user interface.

Similarly, in [27] question topics from natural language are inferred using supervised latent Dirichlet allocation (SLDA) for classification [7]. They focus on questions in Yahoo! Answers and set the number of topics to 50. They discovered that, unsurprisingly, the topic with the lowest probability of remaining unanswered is a seasonal topic (questions were crawled on April 1st and the most answered questions were about “April Fools”). The second and third most answered questions are related to pets and food, respectively.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface</td>
<td>view, image, button, etc.</td>
</tr>
<tr>
<td>Stack Trace</td>
<td>java, error, org, server, etc.</td>
</tr>
<tr>
<td>Large Code Snippet</td>
<td>code, string, new, object, class, etc.</td>
</tr>
<tr>
<td>Web Document</td>
<td>href, page, html, php, etc.</td>
</tr>
</tbody>
</table>

Table 1. Topics and related words (obtained from [25])

4.1.1.2 Length of the Question

In [27] the authors find that the top 10% shortest and the top 10% longest questions have the highest probability of obtaining an answer, while the medium length questions were less likely to be answered. They explain this phenomenon by noting that reading and answering a short question can be accomplished in a very short time. Long questions are mostly expertise-related and need more explanation. These questions attract more users with the same interest and are therefore more appealing to be answered. The authors of this study assume that medium length questions are less interesting and unnecessarily long which makes them less likely to receive an answer. In contrast, according to [6] too short questions are very likely to remain unanswered. Those questions may miss important information, be too vague or unclear. Also,
too time-consuming questions are not very attractive for answerers. In another research on the effect of question length on question quality it was found that for answered questions, the minimum length is 5 characters and the maximum length is 48,258 characters \((M = 1,079; \ SD = 1,389)\); for unanswered questions, these numbers are 19 and 35,588, respectively \((M = 1,300; \ SD = 1,845)\) [21]. Their results show that although unanswered questions have longer length, both answered and unanswered questions, have the same probability of receiving an answer. In the ranking list of importance for differentiating unanswered from answered questions the attribute “question length” gets the same place for both classes. Finally, in [10] the authors found that compared to closed questions, deleted questions had a slightly higher number of characters in the question body. Existing literature thus does not provide a consistent answer to the question of whether and to what extent question length influences question quality. Further, it is not clear whether users mainly look at the length of the question title or the question body in deciding whether to answer the question or not, since question length and question body length are never analyzed separately.

4.1.1.3 Presence of an Example

Providing the audience with an example may help the asker to clarify or specify the information need she is seeking for. However, multiple examples may result in an overload of information and overwhelm the potential answerers making them less willing to answer the question. In programming CQA platforms, users can provide a code example to clarify their information need. Although code snippets are very specific for programming CQA platforms, they can be perceived in a more general way and are therefore included in the current survey. In [24] a few hundred questions were manually analyzed and assigned to ten different categories. Their goal was to investigate which questions are answered well and which remain unanswered. Using qualitative and quantitative data from Stack Overflow, they distinguished ten question categories – how-to, environment, discrepancy, error, decision help, conceptual, review, non-functional, novice and noise. They found that review questions had a high answer ratio as they often contain a code snippet and may have more than one possible good answer. The presence of a code snippet will not only elicit more answers to review questions; also other types of questions may benefit from the extra information provided by a code fragment. According to [6], for example, program specific questions are very hard to answer if no code fragment or detailed explanation is included. In [10] the authors analyze deleted and closed questions. Question that are extremely off topic, have very poor quality or have no activity after a long period of time are deleted. A user can delete her own question when it has not received any answers or upvotes. Closed questions also indicate low quality and are questions which are considered duplicate, subjective, off topic, too localized or not a real question [11]. Similarly to the research of [6], the authors in [11] found that deleted questions had a lower percentage of code blocks compared to closed questions. Interestingly, the presence of a code snippet may have adverse effects as well. A user may not receive an answer if the code is hard to follow or if other users are able to understand the code but cannot see the problem [6]. The possible adverse effects may explain why in [21] the presence of code ranks only ninth in terms of importance for differentiating between answered and unanswered questions.

5 http://stackoverflow.com/help/bounty
6 http://askubuntu.com/tour
7 http://answers.unity3d.com/page/faq.html
5. CONCLUSION

The increasing popularity and usage of Community Question and Answering platforms calls for an investigation into factors influencing content quality and ways to encourage improving the quality. The CQA content can be roughly divided in questions and their answers: in this survey we address the quality of the questions. The number of answers and the question score have been shown to be a good measure of question quality on CQA websites. The question features most frequently used in research on predicting question quality were tags and terms, length of the question, the presence of an example and user reputation. Certain content-related features of the questions have a negative influence on the question quality - questions containing incorrect tags, or that are too localized, subjective or off topic are considered of bad quality. Results on the influence of the question length are more mixed and further research should provide better insights in its importance for the number of answers and the question score. On the other hand, the presence of an example has a positive effect on the question score and the number of answers. Such insights about the effect of these features are useful when engineering forums so as to guide users to improve their questions and to make it more likely they obtain the information they are searching for.

6. REFERENCES


A Framework for Collocation Error Correction in Web Pages and Text Documents

Alan Varghese1, Aparna S. Varde1, Jing Peng1 and Eileen Fitzpatrick2
1. Department of Computer Science
2. Department of Linguistics
Montclair State University, Montclair, NJ, USA
(varghesea2 | vardea | pengj | fitzpatrickc)@montclair.edu

ABSTRACT
Much of the English in text documents today comes from non-native speakers. Web searches are also conducted very often by non-native speakers. Though highly qualified in their respective fields, these speakers could potentially make errors in collocation, e.g., “dark money” and “stock agora” (instead of the more appropriate English expressions “black money” and “stock market” respectively). These may arise due to literal translation from the respective speaker’s native language or other factors. Such errors could cause problems in contexts such as querying over Web pages, correct understanding of text documents and more. This paper proposes a framework called CollOrder to detect such collocation errors and suggest correctly ordered collocated responses for improving the semantics. This framework integrates machine learning approaches with natural language processing techniques, proposing suitable heuristics to provide responses to collocation errors, ranked in the order of correctness. We discuss the proposed framework with algorithms and experimental evaluation in this paper. We claim that it would be useful in semantically enhancing Web querying e.g., financial news, online shopping etc. It would also help in providing automated error correction in machine translated documents and offering assistance to people using ESL tools.

Categories and Subject Descriptors
H.2.8 [Database Management]: Database Applications – data mining H.3.3 [Information Systems]: Information Search and Retrieval - search process, I.2.7 [Artificial Intelligence]: Natural Language Processing - machine translation, text analysis

General Terms
Algorithms, Documentation, Performance, Experimentation, Human Factors, Languages, Verification

Keywords
Web Queries, Automated Translation, ESL Tutors, News Articles, Online Shopping, Machine Learning, NLP

1. INTRODUCTION
We address the problem of collocation errors. These are errors where users enter expressions with words not typically used together, e.g., “powerful tea” and “quick cars”. These do not represent semantically correct collocations in English even though they are syntactically correct. For example, an expression such as “powerful tea” entered in a search for online shopping is not meaningful in correct native-speaker English. However, it is possible that users who are non-native English speakers, on translating from their native language, use such terms in Web queries or text documents. It is also possible that a machine translation from a foreign language document can have the same mistake. To the best of our knowledge, current systems, including automated machine translators, editors such as MS Word and search engines such as Google, do not automatically correct such errors or provide suggestions to users. For example, if a user enters “powerful tea” in a search engine during online shopping, the results contain “powerful” or “tea” or both. However, the user probably means to search for the availability of “strong tea”. It would thus be useful to make the Web search more intelligent by providing correctly collocated responses to such errors and accordingly return semantically appropriate results. Note that from the NLP and Web search angles this is an issue of semantics, not syntax. For example, the term “powerful tea” is syntactically correct as it obeys the rules of English grammar but it is semantically incorrect since it does not convey the intended meaning of the Web query. Likewise such collocation error correction is also directly applicable in the preparation of text documents where an ESL learning tool may be used. Similarly in machine translation such errors can occur and should be corrected to enhance performance.

There has been work on incorporating temporal changes in Web and text data, e.g., [6,7]. There is literature on correlated terms and ontology over the Web, e.g., [11,16]. Soderland et al. [15] deal with discourse analysis and inter-sentential inference generation. All these works, though somewhat relevant to ours, do not address flaws in collocation.

Futagi et al. [5] propose a method for identifying collocation errors using association measures over syntactic patterns. Ramos et al. [14] build an annotation schema with a 3-D topology to classify collocation. Dahlmeier et al. [3] suggests a method of using the native language of the L2 learner to correct collocation errors. Liu et al. [9] propose a probabilistic approach to collocation error correction using British National Corpus (BNC) and Wordnet for language learning tools. These works mainly address collocation for linguistic classification. Some of them require knowledge of the writer’s native language. Also, their focus is not on the semantics of Web queries and text documents, which we deal with in our work.

Park et al. [12] categorizes the different types of collocation errors into insertion, deletion, substitution, and transposition errors. Substitution or alternation errors occur when a non-preferred word is used in place of a more commonly used word. Substitution errors frequently result in collocation errors. In their approach they focus on frequency. While our work can be considered orthogonal to such works, use a variety of similarity measures and combine the measures using machine learning and natural
language processing techniques, in order to provide ranked correct suggestions to users.

The rest of this paper is organized as follows. Section 2 defines in detail the problem of collocation error correction as addressed in this paper. Section 3 describes our proposed framework called CollOrder. Section 4 explains the algorithms we have developed in CollOrder. Section 5 summarizes our evaluation. Section 6 gives the conclusions and future work.

2. PROBLEM DEFINITION

We first introduce the contexts in which collocation errors can occur and then state the precise goals of our work.

2.1 Tools for ESL in Text Documents

Much of the collocation related work that has been done can be applied directly to L2 English learners, when the L2 learner uses an odd collocation. It is useful to highlight such mistakes automatically and provide ranked suggestions. The challenge here is to provide only relevant suggestions. There are two steps to the procedure one is to flag or highlight a mistake and the second to provide relevant ranked suggestions. L1 paraphrasing may be a potential approach in this scenario provided the L1 language of the user of the ESL tool is known. Our goal however, is to develop an approach that can be applied in this situation without the knowledge of the L1 language of the user.

2.2 Automated Machine Translation

There are different papers that address the collocation errors in machine translation. The automated machine translation will most probably create the type of collocation error that a L2 learner makes and even more. In this scenario, L1 paraphrasing can usually be applied since the original language of the document is obviously known. Our goal of proposing a collocation error correction approach can still be applied to this problem without using any knowledge of the original language. It would thus be more generic.

2.3 Querying over Web Pages

Currently, search engines provide alternative suggestions for search expressions in Web queries so that the user can look at them to determine if one of them is better or relevant to their search. This often occurs if the user makes a spelling or grammatical mistake. However, it does not occur for collocation errors as far as we know. We next provide a motivating example to emphasize the importance of collocation error correction in search engines in order to return appropriate Web pages.

Motivating Example: Consider that a user wishes to search for financial news related to black money. If we conduct a Google search using the incorrect collocation “dark money” that could possibly arise due literal translation from the user’s native language, the following Web pages are returned as shown in Figure 1. While the search does provide some information, this is not exactly relevant to the user’s intended query. Now suppose that we provided the user with a suggestion of a correct collocation “black money” and the user searches with this expression thereafter. We clearly see that the resulting Web pages as shown in in Figure 2 are much more applicable to what the user originally intended.

Likewise users can conduct searches on other articles pertaining to financial news and other topics, and it is important to use correctly collocated terms. Another example is “stock market”. Using incorrect collocations such as “stock agora”, “stock mall” would yield undesired Web pages as search results, while using the correct collocation returns more appropriate Web pages including those with financial news on stocks, which would be more helpful. Online shopping is another context in which appropriate search terms are important. We have noticed that using the correct collocation “fast cars” and “strong tea” instead of the collocation errors “quick cars” and “powerful tea” yields better Web pages as responses to user queries, which would help to enhance the online shopping experience.

Figure 1: Example of a collocation error in a Web query for financial news

Figure 2: Web querying for financial news using a correct collocation
2.4 Problem Statement
Considering the various contexts provided here such as machine translation and Web search our goals in this work are twofold:

- Detecting collocation errors in L2 written English in the context of Web queries, machine translation and ESL tools for text documents.
- Providing ranked suggestions as responses to the collocation errors in order to assist users in the respective applications.

The first goal is thus to identify the collocation error in a Web search term or in a sentence in a text document. The next goal pertains to listing all possible corrections to the incorrect collocation and eliminating the improbable ones so that the list of suggestions provided to the user is minimal and as far as possible ordered by relevance in descending order.

3. PROPOSED FRAMEWORK
We propose a framework called CollOrder for detecting collocation errors and suggesting correctly ordered responses to them. This framework is illustrated in Figure 3. It consists of an error detection step followed by an error correction step that includes ranking.

3.1 Error Detection
The input to CollOrder is an expression entered by a user. CollOrder has access to huge text corpora of correct English such as the American National Corpus and the British National Corpus, which we refer to as the “corpus databases”. In the error detection step, CollOrder first performs part-of-speech (POS) tagging, i.e., each term in the expression is assigned a part of speech based on its grammatical category: noun, verb etc. This POS tagging is performed on the assumption that parts of speech of the incorrect collocation match those of the correct collocation. After POS tagging, CollOrder searches for matches, comparing the given tagged terms with those in knowledge bases (KBs) of correct English, such as the American National Corpus, ANC (or the British National Corpus, BNC). If a match for the tagged term is found therein with a frequency greater than system-defined thresholds, the assumption is made that the user has entered a correct expression, e.g., “strong tea”. Thus, the approach would not execute further.

3.2 Error Correction
The error correction step executes if tagged terms in the user expression are not found in corpus databases, implying that a collocation error is detected, e.g., if the user enters “powerful tea”.

3.2.1 Search for Potential Collocates
To correct the error, CollOrder conducts a search for frequently used collocates of each tagged term in the expression, again using system-defined frequency thresholds. So, in this example, it would search for frequent collocates of “powerful” and of “tea”. It could discover several frequent collocates of “tea” such as “strong”, “potent”, “good”, “Indian” etc. These could yield potentially correct responses like “strong tea”, “potent tea”. It could also include frequent collocates of “powerful” such as “drink”, “statement”, “person” and so forth. These could yield several potential responses such as “strong tea”, “Indian tea”, “powerful statement”, “powerful drink” as so forth.

Before proceeding further with the responses to be conveyed as suggestions to the user, we first explain the search process in CollOrder that deserves some attention.

3.2.2 Pre-Compute Collocates for Efficient Search
We have found that detailed searching and parsing over huge databases such as the ANC (with complete statements of correct English) is very time-consuming not feasible to execute recurrently each time a user expression is encountered.

Hence, we propose the following CollOrder Search Heuristic: Instead of an exhaustive search over a huge database, a guided search over a smaller indexed knowledge base containing common collocates of selected parts-of-speech and their collocation frequency is equally effective and more efficient.

Thus, we generate smaller KBs (from corpora such as the ANC) by extracting relevant collocates with frequencies above the threshold. These are called the Collocate Frequency Knowledge Bases (CFKBs) and are far more condensed with only the relevant knowledge. For example the ANC is 10GB while its extracted CFKB is only 400MB. Thus, reduction in size is 86%, proportionately reducing search complexity.

We deploy techniques from [10] in our execution to parse sentences. With reference to our heuristic, we argue that after parsing it is useful to retain only those parts of speech relevant for collocation. From the NLP angle, these are abstracted as:

3.2.2.1 nn: noun compound modifier
A noun compound modifier of an NP is any noun that serves to modify the head noun, e.g., “Oil price futures” nn(futures, oil).

3.2.2.2 advmod: adverbial modifier
An adverbial modifier of a word is a (non-clausal) adverb or adverbial phrase (ADV) that serves to modify a word used in an adverbial sense, e.g., “Genetically modified food” advmod(modified, genetically)

3.2.2.3 amod: adjectival modifier
An adjectival modifier of an NP is an adjectival phrase serving to modify the following noun, e.g., “Sam eats red meat” amod(meat, red)
3.2.2.4 *prt: phrasal verb particle*

The phrasal verb particle relation identifies a phrasal verb, and holds between the verb and its particle. Example: “They shut down the station” *prt*(shut, down).

The knowledge bases hereby generated are several orders of magnitude smaller than the original text corpus databases and contain relevant information for searching of collocates. These can then be used to execute the searches each time a user expression is encountered, which is very efficient.

In order to execute this efficient searching, we implement programs to pre-compute the frequency of collocates and the part-of-speech (POS) tag of collocates found in the original text corpus databases such as ANC. Figure 4 summarizes this approach.

![Figure 4: Creating a CFKB using a corpus DB such as ANC](image)

The first module in Figure 4 deploys the well-known Stanford Parser to parse all the text files in the corpus databases. The American National Corpus is shown here as an example but the same logic applies to other corpora such as the British National Corpus. This module generates a file containing collocates and the POS tag associated with each collocate. The next module in this figure uses the file generated by the first one and creates a relational database containing collocates and the frequency of occurrence along with the POS tag associated with each collocate.

The Collocate Frequency Knowledge base serves as a corpus of collocates in the English Language. Although this is limited to the collocations in the corpus that we use, we argue that ANC and BNC are very comprehensive and the experimental results have proven that it is very relevant. More importantly we would like to highlight one of our contributions to collocation error detection and correction. To the best of our knowledge, this method of pre-computing and materializing collocates in a CFKB accompanied by the Collocate Search Heuristic is unique as an approach for efficient searching of collocates.

Once this efficient searching has been conducted, several potentially correct collocates of terms within the user expression would be discovered. Not all of these would be meaningful in the given context and hence we cannot simply output all of them as suggested responses. It is thus important to conduct filtering and ranking and then convey useful suggestions as the output of CollOrder. We thus proceed to explaining the ranking next. Even though the ranking forms part of the Error Correction step, we are explaining it here as a separate subsection since it encompasses several concepts.

3.3 Ranking the Suggestions

We propose to rank the suggestions using the data mining approach of classification such that it encompasses several measures. The various tasks involved in ranking are as follows.

3.3.1 Pre-filtering the suggestions

Prior to performing any calculations for ranking, CollOrder adopts a pre-filtering approach. It filters out collocates that are close to being antonyms of the input collocate. For example, consider the expression “powerful tea”. A collocate such as “light” could also be discovered for “tea”. However, “light tea” conveys almost the opposite meaning as “powerful tea” and hence should be removed from the list of responses to be suggested to the user. This filtration is conducted by using a synonym database and removing the terms that do not occur therein.

3.3.2 Selecting Measures for Ranking

We propose to deploy the following different measures for ranking as they capture various useful aspects of the data. The respective aspects are explained along with each measure.

3.3.2.1 Conditional Probability

In probability theory the conditional probability of A given B is the probability of A occurring if B is known to occur. It is formulated as:

\[
P(A|B) = \frac{P(A \cap B)}{P(B)}
\]

With reference to our work, this translates into:

\[
\frac{\text{Freq}(A \cap B)}{\text{Freq}(B)}
\]

where N is the total number of words, while A and B are the two terms in the given expression.

Hence, this equates to:

\[
\frac{\text{Freq}(A \cap B)}{\text{Freq}(B)} / \text{N}
\]

For example, consider the suggestion “strong tea” as a response to “powerful tea”. In this context, we calculate the probability of “tea” given “strong”. It is useful to obtain this information because it helps us determine the relative occurrence of the terms. It would be helpful to know how often the term “tea” occurs in conjunction with “strong”, in order to convey it as a suggestion.

3.3.2.2 Jaccard’s Coefficient

We use the Jaccard similarity coefficient as a measure of semantic similarity. It is defined as the size of the intersection divided by the size of the union of the sample sets. The formula is:

\[
J(A, B) = \frac{|A \cap B|}{|A \cup B|}
\]

We calculate Jaccard’s coefficient in CollOrder using the following method. The co-occurrence of two terms \(Freq(A \cap B)\) is calculated by searching through the corpus for the two words with a word window of 32 which we define as co-occurrenceWindow. We refer to the paper Terra et al. [17] to find a suitable window size.

The individual frequencies \(Freq(A)\) and \(Freq(B)\) are obtained by searching through the corpus using a search engine.

Thus, Jaccard’s coefficient is calculated using:

\[
\frac{\text{Freq}(A \cap B)}{\text{Freq}(A) + \text{Freq}(B)}
\]

We consider this measure because it is important to measure this semantic similarity between the term in the user expression and the potential responses to convey suggestions to users. Thus, in using Jaccard’s, we would measure the extent of the semantic similarity between “strong” and “powerful”, “potent and powerful” and so forth, in order to find the appropriateness of
terms such as “strong” and “potent” in being conveyed as the suggested responses “strong tea” and “potent tea” respectively.

### 3.3.2.3 Web Jaccard

Web Jaccard is a semantic similarity measure which is slightly modified form of the Jaccard’s coefficient shown above in that we remove the frequency of intersection of terms from the sum of the individual frequency of occurrences. While searching text documents we propose to count the situations where A and B occur together only once in $F_{a \cup b}$ which is the denominator in the Jaccard coefficient.

$$\text{Web Jaccard}(A, B) = \frac{F_{a \cap b}}{F_{a} + F_{b} - F_{a \cap b}}$$

We calculate the individual terms in the Web Jaccard similar to the manner in which we calculate the Jaccard coefficient and then we apply the above formula to obtain the Web Jaccard. The rationale behind this measure is therefore almost the same as the rationale behind Jaccard’s coefficient with a minor twist for additional robustness, in order to experiment with more measures.

### 3.3.2.4 Frequency Normalized

In addition to other measures, we also consider the fundamental collocation frequency. However, order to conduct ranking based on various measures, it is helpful to normalize the values for frequency since the other measures are all in the range of 0 to 1.

We perform the normalization as follows. From the list of suggested responses, we find the one with the highest frequency and the lowest frequency and consider these as the upper and lower limits for the original range. We then map all values in that range to values between 0 and 1, thereby normalizing them.

This is formulated as follows. For all the expressions considered as suggested responses:

- $\text{Min} = \text{frequency of the lowest occurring collocate}$
- $\text{Max} = \text{frequency of the highest occurring collocate}$

For any given collocate:

$$\text{Frequency Normalized} = \frac{\text{Frequency of collocate} - \text{Min}}{\text{Max} - \text{Min}}$$

### 3.3.2.5 Frequency Ratio

We hereby propose a measure such that its value is normalized between 0 and 1 and such that it has a higher value if the terms co-occur very frequently and a lower value if the terms co-occur less frequently compared to their individual occurrences. This is a measure purely based on frequency of collocation and does not consider similarity with the original expression. For example, if the original user expression is “powerful tea” and a potential suggestion is “strong tea”, the frequency ratio is calculated between “strong” and “tea”. Note the difference between this one and conditional probability. This measures the co-occurrence of both terms while conditional probability measures the occurrence of one term given another. Our rationale for proposing this one is to experiment with a variety of measures including those that do not incorporate the original user expression. The formula for frequency ratio is given as:

$$\text{Frequency Ratio} = \frac{F_{a \cap b}}{F_{a} + F_{b}}$$

### 3.3 Combining the Measures

Empirically, it has been found that different measures yielded good results in different scenarios. We therefore propose to deploy the machine learning paradigm of classification here for combination of measures in order to optimize performance.

We consider each measure as an attribute for the classification task and introduce a target class attribute. We mark all collocations as class $n$ (No) except the ones that are correctly classified a $y$ (Yes). Please refer a sample set given below in Figure 5. This is an example of the training set for the machine learning classifier.

We run a classifier (JRIP) which is an implementation of RIPPER. This classifier implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which was proposed by William W. Cohen as an optimized version of IREP (W.W. Cohen [1])

<table>
<thead>
<tr>
<th>Input</th>
<th>Suggestion</th>
<th>Conditional prob</th>
<th>freqNorm</th>
<th>jaccard</th>
<th>freqRatio</th>
<th>Web Jaccard</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>pure</td>
<td>blue</td>
<td>0.0286207</td>
<td>0.898767</td>
<td>0.002956</td>
<td>0.015862</td>
<td>0.002829525</td>
<td>y</td>
</tr>
<tr>
<td>pure</td>
<td>night</td>
<td>0.009351865</td>
<td>0.607759</td>
<td>0.009523</td>
<td>0.009843</td>
<td>0.0008233</td>
<td>n</td>
</tr>
<tr>
<td>pure</td>
<td>clever</td>
<td>0.001332547</td>
<td>0.092492</td>
<td>0.001777</td>
<td>0.001113</td>
<td>0.001773497</td>
<td>y</td>
</tr>
<tr>
<td>pure</td>
<td>water</td>
<td>0.001397421</td>
<td>0.139483</td>
<td>0.002957</td>
<td>0.001275</td>
<td>0.000268387</td>
<td>n</td>
</tr>
</tbody>
</table>

**Figure 5: Subset of the Training Set for Machine Learning**

We explain the process as follows. We first run the CollOrder program without applying any filtering on a set of incorrect collocates. This yields a comma separated dataset similar to that shown in Figure 5. Each row contains a potential suggestion and also contains all the measures of similarity and frequency and the class is by default n (No). The rows containing correct collocates and the ones that are highly likely to be correct choices are marked as y (Yes). This serves as the training set.

The input to the JRIP algorithm is a comma-separated value file (csv) as shown in Figure 5. The algorithm prunes the rules and produces the following rule which is then used in our program within the ranking section of CollOrder.

$$(\text{Jaccard} >= 0.00338) \text{ AND (frequencyRatio} >= 0.002833))$$

OR

$$(\text{jaccard} >= 0.00177) \text{ AND (frequencyNormalized} >= 0.811321))$$

We have found that this rule has been effective in listing out only the required top-k collocates and thus it automatically determines the value of k, i.e., the number of suggestions to be output for the given user expression (collocation error). So we can assume that Jaccard’s coefficient and Frequency Ratio measures are better than the other measures we considered for this particular situation. Likewise, suitable training sets can be provided for an exhaustive list of common collocates in other situations and the system would learn the rules that are required for ranking the collocates. Upon running this procedure in CollOrder, we have obtained an effective ranking for a variety of examples that yielded good results experimentally. We have considered various user expressions that are incorrect collocations and conducted the ranking of suggested responses using the procedure described herein. This output of CollOrder including the correct responses given to users along with the ranking in various scenarios has been evaluated as effective by detailed user surveys. This is further corroborated in our section on evaluation.

To the best of our knowledge, such an approach of deploying a classifier to combine measures of similarity in the context of collocation error correction has not been used in the literature. Hence we claim this to be a unique contribution on our part.
Note that throughout in the explanation of CollOrder we have explained the logic in our approach using two terms in a collocation. However, the same logic holds good for three terms. For example if the user enters an expression such as “incomplete money help”, instead of the more appropriate “partial financial aid” the terms “incomplete money” and “money help” would both be considered separately and the approach would proceed in the same manner to provide suggestions.

In the rare case that no correct collocation is found within threshold in our CFKB, the system indicates that a suitable response is not obtained from our knowledge bases. This is an issue of sparse data to be addressed as future work.

4. ALGORITHMS

We now proceed with outlining the algorithms that we developed in the CollOrder approach. Figure 6 gives the details of the CollOrder approach based on the explanation given in Section 3.

4.1 Pre-computing Collocates in CollOrder

The algorithm for pre-computing collocates is given below as Algorithm 1. This algorithm determines the “correct collocates” and creates the CFKB. We run the Stanford parser on each of the text documents in the ANC. We find the tagged items present in each of the sentences. If the “Part of Speech Tag” is among those we decided to consider then we store the tagged item as a “collocate” in the CFKB. The function “storeinCFKB” increments the frequency by 1 if it already exists or else it stores it with a frequency of 1.

```plaintext
for all document ∈ ANC do
  sentences ← parse(document)
  for all sentence ∈ sentences do
    for all tagged item ∈ sentence do
      if POSTAG ∈ ('amod', 'advmod', 'nn', 'prt') then
        storeinCFKB(taggedItem)
      end if
    end for
  end for
end for
```

Algorithm 1: Pre-computing of Collocates in CollOrder

4.2 CollOrder Main Routine

The main algorithm or routine that invokes the whole CollOrder approach is given next as Algorithm 2. The input to CollOrder is a sentence. First the sentence is parsed to find the tagged items in the input. If the tagged item has a POS tag that we are considering then we check in our CFKB to see if it is present with a frequency greater than threshold (we use 10 as the threshold). If not present then we call our error correction routine. Although we have explained this with a sentence, the same logic applies for a search expression in a Web query or any expression / sentence in machine translation.

```plaintext
function COLLORDER(sentence)
  taggedItems ← parse(sentence)
  for all item ∈ taggedItems do
    if posTag ∈ ('amod', 'advmod', 'nn', 'prt') then
      cfkItem ← findInCFKB(item)  
      if cfkItem = null then
        ERRORCORRECTION(item)
      end if
    end if
  end for
end function
```

Algorithm 2: CollOrder Main Routine

4.3 Error Correction in CollOrder

The following algorithm, namely Algorithm 3, gives the details of error correction in the CollOrder approach.

In this algorithm, the ERRORCORRECTION procedure:

- Searches through the CFKB for potential suggestions.
- Filters out antonyms
- Mark the suggestions that are synonyms of the input collocate
- Calls the CALCULATE MEASURES routine.
- Calls the FILTER BY JRIP RULES routine to filter out the results by applying the JRIP rules that we extracted to the parameters we calculated in the CALCULATE measures routine
- Prints out the top-k results in the order of correctness of collocation based on the ranking

Figure 6: Details of the CollOrder Framework
Having discussed these algorithms we developed in CollOrder, we now proceed with the details of our experimental evaluation.

Algorithm 3: Error Correction in CollOrder

5. PERFORMANCE EVALUATION

We have developed a software tool based on our CollOrder framework. This tool accepts any expression as an input from the user and if a collocation error is detected it provides a list of suggestions to the user ranked in the order of correctness. To evaluate the effectiveness of this tool and hence our CollOrder framework for detection and correction of collocation errors, we have conducted a performance evaluation using Mechanical Turk as described next.

5.1 Evaluation using Mechanical Turk

The evaluation was performed using Amazon mechanical Turk. 20 screen shots of the CollOrder web application was provided to the evaluators. Each evaluation is called a Human Intelligence Task or HIT and it would be presented like what is shown below.

Amazon Turk allows us to download the results as a csv which gives us the url of the image that we asked the user to evaluate and the columns with the two entries that the users provided.

Out of the 225 reviews done we find that 92.44% of the time the users agreed that a correct alternative to the collocation errors, i.e., odd sounding collocations, was suggested. This indicates that the error correction step is indeed functioning well.

5.2 Effectiveness of CollOrder in Web Search

Let us assess the usefulness of CollOrder with respect to search results in Web pages. Consider a user evaluation example where a non-native speaker incorrectly enters an expression such as “quick car” in a Web query, when they really meant to search for “fast cars”. This could for example occur in the context of online shopping. The response given by a search engine for this expression is shown in Figure 9. We have 1,270,000 results which do not seem very relevant to the user’s really intended query.

Consider the response given by the CollOrder framework in this situation with the user expression “quick car” which is detected as a collocation error.

We notice that CollOrder outputs only one suggestion here as shown in Figure 8 based on the execution of the search and ranking. Consider that the user conducts the Web search with the suggestion “fast car” as shown in Figure 8 based on the execution of the search and ranking. Consider that the user conducts the Web search with the suggestion “fast car” given by CollOrder as a correct collocation.

We notice the following search results as shown in Figure 10. We have 10,800,000 results and the results are much more relevant. In fact, these also include images for fast cars. The Google suggestions are also more relevant to the intended search.
Likewise, we find that the suggestions given by the CollOrder framework can indeed improve the effectiveness of query results in Web pages. These can be very useful in applications such as online shopping.

While we have evaluated the effectiveness of CollOrder in the context of Web querying and its applications, similar justification can be applied in other contexts such as ESL tools for text documents and automated machine translation. We briefly explain these below.

5.3 Potential Use in Machine Translation

A full consideration of the use CollOrder in machine translation is beyond the scope of our current paper. However, we present the following as potential effectiveness of the framework.

Machine translation software can convert a sentence from a foreign language to English. This sentence can then be sent to CollOrder. The CollOrder framework can determine whether the
collocations in the sentence are correct. If they are incorrect, it can provide ranked suggestions as correct collocations. The machine translation software can then decide whether the sentence needs to be changed. If so, it can either change the collocation to the suggested collocations, or else flag it for manual review with a list of ranked suggestions.

5.4 Potential Use in ESL Writing Aids

The CollOrder framework can also assist in evaluating text written by non-native speakers such that it can be useful in ESL writing aids. Again, this has not been implemented but is considered as potential effectiveness of CollOrder. The process of text evaluation in the writing aid could be as shown in Figure 11.

The software can read each sentence and invoke CollOrder for each sentence. If CollOrder marks the collocations as odd then the suggestions can be used to correct the text or provide a suggestion list for manual review.

![Figure 11: Text evaluation flowchart](Image)

5.5 Discussion on Further Challenges

Based on the performance evaluation of CollOrder, it is clear that effective responses to collocation errors are provided. However, there is scope for more improvement. In this context, we outline further challenges as discussed below.

Domain Knowledge: A significant challenge includes capturing knowledge on domain-specific aspects for several text corpora, e.g., those in scientific fields. Such knowledge may not be present in standard databases such as the American National Corpus, as a result of which we may face the problem of not finding any correct response to a given expression. In order to address this we propose to include several other domain-specific databases in our search besides standard corpus databases of correct English sentences, which in turn brings us to another challenge.

Literary Allusion: It is challenging to define a precise notion of correctness to clearly distinguish between acceptable and unacceptable terms in collocation, considering issues such as literary allusion, e.g., “inanimate human”. If a writer has used such an expression, there seems to be no appropriate response here. If this term appears in the database of correct English it could possibly be a poetic term, although with respect to common parlance it should be detected as an error. Thus the notion of correctness needs to take into account such anomalies. If we have domain-specific databases on various literary subjects then this challenge is even more prominent.

Sparse Data: The issue of correct but sparse data poses an issue, which could also lead to the problem described earlier of not finding any correct responses within a given threshold. Some terms may not show up in text corpora even though they are correct expressions. It is therefore important to address this issue and propose appropriate solutions.

These and other such challenges provide the potential for further research in the area of collocation error correction.

6. CONCLUSIONS

In this work, we have proposed a framework called CollOrder to detect and correct collocation errors in written English. This is useful in the context of searches in Web pages, ESL help in text documents and also in automated machine translation. The main contributions of this research include:

- Proposing the overall CollOrder framework based on an integrated approach incorporating natural language processing and machine learning
- Proposing efficient searching in CollOrder through search heuristics and pre-computation of collocates with materialization
- Proposing a method to rank collocates in CollOrder based on classification and similarity measures
- Implementing the CollOrder framework using real data and developing a GUI for user interaction
- Conducting detailed user surveys and objectively evaluating the effectiveness of CollOrder
- Providing inputs as correct collocations in various contexts such as Web queries, ESL tools and machine translation to assist L2 users

This work would be useful to developers of intelligent tutoring systems, search engines, text editors and translation tools. It would be of interest to the database, data mining and AI communities due to emphasis on aspects such as search and ranking, natural language processing, machine learning and Web data. It would also appeal to users from various domains such as finance, marketing and linguistics.

Future work includes addressing challenges such as sparse but correct data, literary allusion and domain knowledge. There is potential for further enhancing the performance of the CollOrder framework by using powerful computers and an in-network synonym and antonym dictionary. We could also host the search engine for the ANC /BNC in-house on a powerful server. This would make the framework more efficient thus providing even better user satisfaction.

7. REFERENCES


Theoretical Foundations and Algorithms for Outlier Ensembles

Charu C. Aggarwal
IBM T. J. Watson Research Center
Yorktown Heights, NY, USA
charu@us.ibm.com

Saket Sathe
IBM Research - Australia
Melbourne, Australia
ssathe@au.ibm.com

ABSTRACT

Ensemble analysis has recently been studied in the context of the outlier detection problem. In this paper, we investigate the theoretical underpinnings of outlier ensemble analysis. In spite of the significant differences between the classification and the outlier analysis problems, we show that the theoretical underpinnings between the two problems are actually quite similar in terms of the bias-variance trade-off. We explain the existing algorithms within this traditional framework, and clarify misconceptions about the reasoning underpinning these methods. We propose more effective variants of subsampling and feature bagging. We also discuss the impact of the combination function and discuss the specific trade-offs of the average and maximization functions. We use these insights to propose new combination functions that are robust in many settings.

1. INTRODUCTION

The problem of outlier ensembles has recently received increasing attention in the research community [1; 2]. Ensemble analysis is used extensively for high-dimensional outlier detection [3; 12; 13; 14; 18]. In high-dimensional outlier detection, multiple subspaces of the data are explored in order to discover outliers. One of the earliest formalizations [14] of outlier ensemble analysis is based on high-dimensional outlier detection. Other recent methods for ensemble analysis in outlier detection are discussed in [11; 15; 17; 19; 21; 24]. Outlier detection is an unsupervised problem, in which labels are not available with data records. As a result, it is generally more challenging to design ensemble analysis algorithms. For example, in the case of boosting, the classifier algorithm needs to be evaluated in the intermediate steps of the algorithm. Such methods are generally not possible in the case of outlier analysis. As discussed in [2], there are unique reasons for ensemble analysis to be generally more difficult in the case of outlier analysis, as compared to classification. In spite of the unsupervised nature of outlier ensemble analysis, we show that the theoretical foundations of outlier analysis and classification are surprisingly similar. Several arguments have been recently proposed on the theory behind outlier ensembles. In some cases, incorrect new explanations are proposed to explain experimental results, which can be explained by old and well-known ideas. Such a confusion is an impediment to the proper development of ideas in the field because future papers would likely try to explain ensemble improvements in a similar way; this could cause even further confusion. It is also particularly important to give proper attribution and credit to the well-known ideas that explain these results. Our work establishes a correct theoretical understanding of outlier ensemble analysis in terms of well-known ideas from classification. We will also show how these theoretical results can be leveraged to design several new ensemble algorithms.

This paper is organized as follows. In the next section, we provide a review of the bias-variance trade-off for outlier detection, and its similarity and differences with the corresponding trade-off in classification. The applications of these theoretical foundations are discussed in section 3. Section 4 discusses the application of the theoretical foundations to the bias-variance tradeoff. The experimental results are discussed in section 5. Section 6 discusses the conclusions and summary.

2. THE BIAS-VARIANCE TRADEOFF FOR OUTLIER DETECTION

The bias-variance tradeoff is often used in the context of supervised learning. Although it might seem at first sight that labels are required to quantify the bias-variance tradeoff, it turns out that this quantification is also applicable to unsupervised problems, simply by treating the dependent variable as unobserved. Most outlier detection algorithms output scores to quantify the “outlierness” of data points. After the scores have been determined, they can be converted to binary labels. All data points with scores larger than a user-defined threshold are declared outliers. An important observation about outlier scores is that they are relative. In other words, if all scores are multiplied by the same positive quantity, or translated by the same amount, it does not change various metrics (e.g., receiver operating characteristic curves (ROC)) of the outlier detector, which depend only on the ranks of the scores. This creates a challenge in quantifying the bias-variance tradeoff for outlier analysis because the uniqueness of the score-based output is lost. This is because the ROC provides only an incomplete interpretation of the scores (in terms of relative ranks). It is possible to work with crisper definitions of the scores which allow the use of more conventional error measures. One such approach, which preserves uniqueness of scores, is that the outlier detectors always output standardized scores with zero mean, unit variance, and a crisp probabilistic interpretation. Note that one can always
apply [2] a standardization step as a post-processing phase to any outlier detector without affecting the ROC; this also has a natural probabilistic interpretation (discussed below). Consider a data instance denoted by $X_i$, for which the outlier score is modeled using the training data $D$. We can assume that an ideal outlier score $y_i$ exists for this data point, even though it is unobserved. The ideal score is output by an unknown function $f(X_i)$, and it is assumed that the scores, which are output by this ideal function, also satisfy the zero mean and unit variance assumption over all possible points generated by the base data distribution:

$$y_i = f(X_i)$$

(1)

The interpretation of the score $y_i$ is that by applying the (cumulative) standard normal distribution function to $y_i$, we obtain the relative outlier rank of $X_i$ with respect to all possible points generated by the base data distribution. In a sense, this crisp definition directly maps the score $y_i$ to its (percentile) outlier rank in $(0, 1)$. Of course, in practice, most outlier detection algorithms rarely output scores exactly satisfying this property even after standardization. In this sense, $f(X_i)$ is an oracle that cannot be computed in practice; furthermore, in unsupervised problems, we do not have any examples of the output of this oracle.

This score $y_i$ can be viewed as the analog to a numeric class variable in classification/regression modeling. In problems like classification, we add an additional term to the RHS of Equation 1 corresponding to the intrinsic noise in the dependent variable. However, unlike classification, where the value of $y_i$ is a part of the observed data for training points, the value $y_i$ in unsupervised problems only represents a theoretically ideal value (obtained from an oracle) which is unobserved. Therefore, in unsupervised problems, the labeling noise’ no longer remains relevant, although including it makes little difference to the underlying conclusions.

Since the true model $f(\cdot)$ is unknown, the outlier score of a test point $X_i$ can only be estimated with the use of an outlier detection model $g(X_i, D)$ using base data set $D$. The model $g(X_i, D)$ is only a way of approximating the unknown function $f(X_i)$, and it is typically computed algorithmically. For example, in $k$-nearest neighbor outlier detectors, the function $g(X_i, D)$ is defined as follows:

$$g(X_i, D) = \alpha \text{KNN-distance}(X_i, D) + \beta$$

(2)

Here, $\alpha$ and $\beta$ are constants which are needed to standardize the scores to zero mean and unit variance. It is important to note that the $k$-nearest neighbor distance, $\alpha$, and $\beta$ depend on the specific data set $D$ at hand. This is the reason that the data set $D$ is included as an argument of $g(X_i, D)$.

If the function $g(X_i, D)$ does not properly model the true oracle $f(X_i)$, then this will result in errors. This is referred to as model bias and it is directly analogous to the model bias used in classification. For example, the use of $k$-NN algorithm as $g(X_i, D)$, or a specific choice of the parameter $k$, might result in the user model deviating significantly from the true function $f(X_i)$ . A second source of error is the variance. The variance is caused by the fact that the outlier score directly depends on the data set $D$ at hand. Any

If there are errors in the feature values, this will also be reflected in the hypothetically ideal (but unobserved) outlier scores. For example, if a measurement error causes an outlier, rather than an application-specific reason, this will also be reflected in the ideal but unobserved scores.

data set is finite, and even if the expected value of $g(X_i, D)$ correctly reflects $f(X_i)$, the estimation of $g(X_i, D)$ with limited data would likely not be exactly correct. If the data set $D$ is relatively small, there will be a variance in the estimation of $g(X_i, D)$, which is significant. In other words, $g(X_i, D)$ will not be the same as $E[g(X_i, D)]$ over the space of various random choices of training data sets $D$. This phenomenon is also sometimes referred to as overfitting. The model variance is high when the same point receives very different scores across different choices of training data sets. Although one typically does not distinguish between training and test points in unsupervised problems, one can easily do so by cleanly separating the points used for model building, and the points used for scoring. For example, a $k$-NN detector would determine the $k$ closest points in the training data for any point $X_i$ in the test data. We choose to demarcate training and test data because it makes our analysis cleaner, simpler, and more similar to that of classification; however, it does not change the basic conclusions.

Let $D$ be the training data, and $X_1, \ldots, X_n$ be a set of test points whose (hypothetically ideal but unobserved) outlier scores are $y_1, \ldots, y_n$. We use an unsupervised outlier detection algorithm that uses the function $g(\cdot, \cdot)$ to estimate these scores. Therefore, the resulting scores of $X_1, \ldots, X_n$ using the training data $D$ are $g(X_1, D) \ldots g(X_n, D)$, respectively. The mean-squared error, or MSE, of the detectors of the test points over a particular realization $D$ of the training data is:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - g(X_i, D))^2$$

(3)

The expected MSE, over different realizations of the training data, generated using some random process, is as follows:

$$E[\text{MSE}] = \frac{1}{n} \sum_{i=1}^{n} E[(y_i - g(X_i, D))^2]$$

(4)

The different realizations of the training data $D$ can be constructed using any crisply defined random process. For example, one might construct each instantiation of $D$ by starting with a larger base data set $D_0$ and use random subsets of points, dimensions, and so on. The term in the bracket on the RHS can be re-written as follows:

$$E[\text{MSE}] = \frac{1}{n} \sum_{i=1}^{n} E[(y_i - f(X_i)) + (f(X_i) - g(X_i, D))^2]$$

(5)

Note that we can set $y_i - f(X_i)$ on the RHS of aforementioned equation to 0 because of Equation 1. Therefore, the following can be shown:

$$E[\text{MSE}] = \frac{1}{n} \sum_{i=1}^{n} E[(f(X_i) - g(X_i, D))^2]$$

(6)

This RHS can be further decomposed by adding and subtracting $E[g(X_i, D)]$ within the squared term:

$$E[\text{MSE}] = \frac{1}{n} \sum_{i=1}^{n} E[(f(X_i) - E[g(X_i, D)])^2] +$$

$$+ \frac{2}{n} \sum_{i=1}^{n} E[(f(X_i) - E[g(X_i, D)]) (E[g(X_i, D)] - E[g(X_i, D)])] +$$

$$+ \frac{1}{n} \sum_{i=1}^{n} E[(E[g(X_i, D)] - (X_i, D))^2]$$

SIGKDD Explorations Volume 17, Issue 1 Page 25
3. LEVERAGING BIAS-VARIANCE IN OUTLIER ENSEMBLES

The similarity in the theoretical underpinnings of classification and outlier analysis is very convenient. As long as an ensemble method in classification does not require knowledge of the class labels, it can be extended relatively easily to outlier detection.

3.1 Extending Bagging to Outlier Detection

Bagging is used commonly in classification to reduce variance. Typically, a bootstrapped sample (i.e., sample with replacement) is drawn in order to construct the training data. The predicted value of the test point is averaged over multiple training samples because the averaged prediction has lower variance. Although it is possible to use bagging for outlier detection, the main problem with doing so is that many base detectors like LOF are not very robust to the presence of repeated points, which increases bias. In some variants of bagging for classification, subsampling is used instead of bootstrapping [6; 7; 8; 20]. In this variant of bagging methods, bootstrapping is not used. Rather, training samples are selected from the data without replacement. The prediction of each test point is computed by constructing a model on each subsample, and then averaging the prediction from various subsamples. This variant is referred to as subagging or subsampling [6; 7; 8; 20]. As in the case of bagging it has been shown [6; 7; 8] that the primary effect of subagging is to reduce the variance. Even though subagging is less popular than bagging, it has been shown that subagging is virtually equivalent to bagging and might even have accuracy and computational advantages under many circumstances [6; 7; 8].

The subsampling (subagging) approach can also be generalized directly to outlier detection. Each point in the data is scored with respect to the subsample by a base outlier detector, whether the point is included in the subsample or not. The scores across different subsamples are then averaged. Recently, this adaptation has been explored for outlier detection [24]. Unfortunately, however, this work does not clarify the direct adaptation from the classification domain, and instead provides a different (and incorrect) theoretical explanation.

3.2 Prevailing Misconceptions on Subsampling

Like classification, the subsampling (subagging) approach can be simply explained with the use of the bias-variance trade-off, by treating the dependent variable as unobserved in the unsupervised setting. However, in an attempt to create new theory for outlier ensembles, it has been stated in [24], that the unnormalized k-NN-distances in d-dimensional data set increase proportionally to (k/n1)1/d for a uniformly distributed outlier region containing n1 points, and the distances increase proportionally to (k/n2)1/d for an inlier region of the same size containing n2 > n1 points (in expectation). It is claimed that the absolute outlier-inlier gap (k/n1)1/d − (k/n2)1/d increases if we reduce both n1 and n2 by the same factor f < 1 via subsampling. Specifically, the multiplicative factor by which the gap increases is (1/f)1/d. It has been claimed that such an increase in contrast makes the inversion in scores between outliers and inliers less likely. Henceforth, we refer to this argument as the “outlier-inlier inversion argument.”

This is, however, an incorrect argument. It is important to understand that downsampling increases the absolute value of the kNN distances (i.e., scales up the scores) because...
of greater sparsity of the data. Therefore, if one used the kNN distances as proxies for the outlier scores, then the score differences between the outliers and the inliers will also proportionately increase. This has no direct impact on the effectiveness of the outlier detection algorithms, because it is merely a scaling issue of the scores. For example, if one multiplied all the outlier scores by \( C > 1 \), then the absolute divergence between the outliers and inliers will increase, but there will no impact on performance metrics of outlier detection algorithms, such as its receiver operating characteristic. The scenario with subsampling is similar because all expected KNN scores are scaled up in the sparsified subsample by a constant factor of \( 1/f^{1/d} \). It is important to understand that the absolute divergence of the scores between outliers and inliers has no significance unless it is properly compared to the effect on the variance of the scores resulting from this approach. Variance is a key factor regulating the rank-wise correctness of the scores. Variances are scaled up proportionately to \( C^2 \), when scores\(^2\) are scaled up by a factor of \( C \). Larger variances make inversion more likely. As we will show in some experimental results in Appendix A, the theoretical claims of “outlier-inlier inversion” are not backed up even over data sets, approximately satisfying the locally uniform assumptions in [24] under which the theoretical results are derived. The inversion argument is quite loosely argued, because it is claimed only for unnormalized k-NN distances in lieu of probability densities; scaling/subsample size impacts the former but not the latter. It does not explain improvements for subsampling in general, or the fact that the experimental improvements in [24] are obtained with the use of distance-normalized algorithms like LOF. In fact, as we will see later, LOF-like algorithms show much larger ensemble-based improvements as compared to unnormalized algorithms. This behavior is consistent with the bias-variance explanation for outlier ensembles, similar to that in classification.

The paper [24] starts by making the (correct) observation that subsampling [with averaging] reduces the randomness “as expected.” This can perhaps be viewed as an informal understanding of variance reduction, which is fairly obvious in such settings because of the earlier subsampling results in the classification domain [6; 7; 8]; even the experimental results in [24] use a classification framework. However, the work in [24] does not try to formally relate to or even cite the existing subsampling results in the classification domain. In fact, the paper explicitly discounts the similarity with the classification problem as a “generic” and “loosely argued” view that does not explain all the performance gains, and it argues for the need for alternative theoretical models in outlier ensembles to the bias-variance models popularly used in classification. The result of this alternative analysis is that it does not properly model the bias component, which has a strong impact on the results in the paper. In this context, the paper [24] goes on to make a very surprising (incorrect) statement which seems to support the “outlier-inlier inversion argument”: “Another, more interesting reason for the improved performance is that the base method applied to a smaller subsample of the whole data often shows an improved outlier detection rate, as compared to the same method applied on the whole data set.” In other words, the statement claims that one can often expect to perform better outlier detection by randomly throwing\(^3\) away a majority of the data in the model building phase! Note that this is a statement about the performance of a single detector rather than the ensemble, and a set of box-plot figures on the performance of component detectors are also shown to experimentally support this argument in [24]. It is often tempting for researchers to simply accept such engagingly counterintuitive statements without question; however, in this case, this absurd statement is contrary to the most basic principles of statistics. The “less-data-is-better” argument seems almost magical, and it disagrees with everything we know about data science. When speaking of the performance of individual ensemble components (i.e., base detectors of ensemble), one cannot even fairly compare the subsampled performance of the algorithm with that on the original data set, if the parameters of the algorithm are fixed. For example, if we used a k-NN algorithm with \( k = 90 \) on a data set with 1000 points, then one must use \( k = 9 \) for a 10% subsample of size 100 points (i.e., retain same percentile value for \( k \)) to ensure that the subsampled algorithm does not have very different bias characteristics. If the value of \( k \) is fixed across different subsample sizes, then the specific quirks (i.e., bias) of the detector on a particular data distribution will dominate the performance. In fact, for different choices of \( k \) on the same data set and algorithm, the change in bias caused by subsampling could either help or hurt the base detector. The paper [24] only shows experimental scenarios in which the bias component helps the base detector. As a result, an incomplete picture is provided about the effectiveness of subsampling. We can already see that omitting the bias component in any theoretical analysis leads to an incomplete understanding of the effectiveness of subsampling. Although subsampling can improve the accuracy of outlier detectors in general, the reasons for doing so follow trivially from the known results on subsampling [6; 7; 8] in the classification setting, and these are the only valid arguments.

Effects of Bias

It is noteworthy that if we use random draws of data sets with a particular data size, then the bias of a particular algorithm will depend on the size of the subsample being drawn. A different way of understanding this is that if we apply Equation 7 to only the universe of data sets of a particular size \( S \), the bias term will be sensitive to the value of \( S \). Relative to the full data set, the accuracy can be improved or worsened, depending on whether the bias is increased or reduced. The effect is, of course, highly data distribution-, algorithm-, and parameter-specific. In fact, the improved performance of the individual detectors in [24] (see Figures 4–7 of that paper), is entirely an artifact of this bias but for other data sets/algorithms/parameters, the results could be different. On the other hand, the variance term in Equation 7 will almost always increase with smaller subsamples (i.e., smaller \( S \)) because of the statistical unreliability of using less data.

\(^2\)When a random variable is scaled by a factor of \( a > 1 \), its variance is scaled up by \( a^2 \).

\(^3\)In subsampling, only the sampled portion of the data is used for model building, although all points are scored against the model.

\(^4\)This is only an approximate adjustment. For some algorithms like LOF, the adjustment becomes even more approximate.
which a $k$-NN algorithm shows improved performance with increasing values of $k$. In this case, the size of the sampled data set is important; if one fixed the value of $k$, and downsamples the data by a factor of $f < 1$, one has effectively increased the percentile value of $k$ by a factor of $1/f$. Therefore, if you used a 9-NN algorithm on a sample of 100 points, the bias would be similar to a 90-NN algorithm on a sample of 1000 points, and it would not be comparable to the bias of a 9-NN algorithm on a sample of 1000 points. In data sets, where the accuracy of a $k$-NN algorithm increases with $k$ on the full data set, subsampling with fixed $k$ will generally improve the accuracy of an individual detector on a single subsample. Even though reduced subsample size has a tendency to reduce accuracy because of increased variance, the accuracy can increase when the bias effects in a particular data set are sufficiently large. On the other hand, in data sets, where the accuracy of a $k$-NN algorithm reduces with $k$ on the full data set, subsampling with fixed $k$ will generally have significantly reduced accuracy of individual detectors because of the double whammy of greater bias and variance from smaller subsample size. In general, it is not necessary for a data set to show a monotonic trend with increasing values of $k$, in which case the bias is entirely unpredictable and completely dependent on the value of $k$ selected for the base method. Therefore, no general statement can be made about the base detectors, although the ensemble performance might improve because of the reduced variance of the ensemble combination; this is not a new argument [6; 7; 8]. The aforementioned observations for unnormalized kNN-distances are also roughly true for LOF-like algorithms, but more approximately so. The improved box-plot performance of component detectors in [24] at smaller subsample sizes (see Figures 4–7 of that paper), can be largely attributed to the choice of the parameter $k$ and data sets used.

In order to show this effect, we performed simulations with a number of real data sets with varying accuracy trends with $k$ (described in detail in section 5.1). In this approach, the average distance to the $k$-nearest neighbor distances [4] is reported as the outlier score. We first used the unnormalized distances because the inversion is theoretically claimed [24] for unnormalized distances. Furthermore, adjusting the value of $k$ for bias is easier in this case than in the case of LOF, although they are roughly true in the latter case. The data sets with increased accuracy with increasing values of $k$ are shown in Figure 1, and the data sets with reduced accuracy with increasing values of $k$ are shown in Figure 2. We reported the Area Under Curve (AUC) of Receiver Operating Characteristics (ROC) curves. Each row contains three figures for a single data set. The leftmost figure of each row shows the performance of the full data set with increasing values of $k$. The middle figure of each row shows the performance of the subsample with fixed values of $k$, but varying subsample size $n_i$. In the rightmost figure of each row, we adjusted the value of $k$ proportionally to subsample size with the formula $k_i = \lceil k_0 \cdot (n_i/n_0) \rceil$, where $n_0$ was the size of the
full data set and \( k = k_0 \) was used for the full data set. The value of \( k_0 \) in the rightmost figure was always selected to be 10 times the fixed value of \( k \) in the middle figure. As a result, the same\(^5\) value of \( k \) was used at subsampling rates of 0.1 in both the fixed-\( k \) and adjusted-\( k \) cases. However, the performance on the full data would be very different in these cases because of a value of \( k \), which is different by a factor of 10. We ran the base detector 100 times with randomly chosen subsamples, and report the box plots, which show the median (blue line in middle of box) and mean (red dot) performances of the component detectors. Note that we are only focusing on component detector performance here in order to understand the bias effects. It is understood that the ensemble will perform better because of known variance reduction effects of subsampling [6; 7; 8]. Nevertheless, we will show in a later section that the performance of component detectors do affect the final ensemble performance to a large extent.

It is evident that for all data sets with increasing accuracy with \( k \), reduction of subsample size improved the performance of the base detector (Figure 1(b), (e), (g)), when the value of \( k \) was fixed across different subsample sizes. On the other hand, for data sets with reducing accuracy with increasing value of \( k \), the performance was drastically reduced (Figure 2(b), (e), (g)) by reducing subsample size. In other words, exactly opposite trends were obtained in the two types of data sets represented by Figures 1 and 2, respectively.

The most interesting results were for the case where an adjusted value of \( k = [k_0 \cdot (n_i/n_0)] \) was used. In these cases, the bias effects have been largely removed, and one can see only the impact of the variance. In this case, consistent trends were observed in the two types of data sets. In most cases, the accuracy reduced (modestly) with smaller subsample sizes, in both types of data sets (Figures 1(c), (f), (i), and Figure 2(c), (f), (i)). This suggests that contrary to the counter-intuitive results suggested in [24], smaller subsamples provide worse performance because of increased variance, once the data-dependent bias component has been removed. It is noteworthy that if the optimal value of \( k_0 \) on the full data set is less than \( n_0/n_i \), then subsampling with \( n_i \) points has an inherent disadvantage for the component detectors, because there is no way of simulating this bias performance on the subsample at any adjusted value of \( k \geq 1 \). This is a simple artifact of the fact that randomly throwing away data leads to irretrievable loss in ability to represent the underlying distribution accurately for outlier detection. In some data sets, such as the Lymphography data set, we found that the behavior of the algorithm with increasing values of \( k \) was algorithm dependent (e.g., Avg-KNN versus LOF-like algorithms). The results are shown in Figure 3(a). The corresponding behavior of the component detectors in subsampling mirrored this behavior. For example, by fixing \( k = 2 \), the Avg-KNN detector (Figure 3(b)) showed oppo-

---

\(^5\)The (roughly similar) boxplots show random variations.
Figure 3: Effects of using different algorithms and parameter settings. The bias is highly dependent on the choice of algorithm and parameter setting. However, given the ground-truth, it is easy to predict by plotting AUC versus k.

The ground-truth AUC performance versus k on the full data set. Of course, since we do not have the ground-truth available in unsupervised problems like outlier detection, there is no way of practically making use of this fact in real settings. The sensitivity of the base detectors to the subsample size also has an important impact on the ensemble performance. As we will show in a later section, even the ensemble performance can be worse than the base detector in some cases. This is because a significant part of the improvements in [24] can be attributed to the better performance of the base detectors at lower subsample sizes. However, since the improvements of the base detector with reducing subsample size, as shown in [24], are unpredictable, one cannot bank on it to improve the final ensemble performance of subsampling in every case. In fact, this unpredictable effect, when adverse, can and will swamp the ensemble performance. The main reason that base detectors improve at lower subsample sizes in [24] is not because of the truth of the “outlier-inlier inversion hypothesis” in the base detectors. Rather, the chosen value of k for the base detectors was always around 10% of a near-optimal value on the full data set, and the performance difference between these two values of k on the full data was very large. While discussing parameter choices of various data sets, the authors do state that the value of k is sensitive to the original data set size; yet they do not adjust the value of k for subsampled components. The sensitivity of k to data size was used only as a justification for setting k to the larger value of 50 in the Satimage-2 data set because of reachability smoothing and the quirky harmonic normalization.

---

We used a similar preprocessing phase as in [24] for Satimage-2, which involved sampling one of the classes. The results do vary significantly across different samples and are therefore not exactly comparable to those in [24].

The prediction is even rougher for LOF because of reachability smoothing and the quirky harmonic normalization.
of its large size. All other data sets, including an even larger synthetic data set, were tested at suboptimally small values of $k = 2$ or $3$.

Data sets in which pareto-extremes represent outliers often show improved accuracy with increasing values of $k$. The simplest example is a single Gaussian distribution in which the tails are viewed as outliers. We generated a standard normal distribution of 2000 points where the 5% of points furthest from the mean were tagged as outliers. A plot of the AUC versus $k$ for both the Avg-KNN algorithm and the LOF algorithm is shown in Figure 4. It is evident that the AUC increases rapidly with $k$ and stabilizes quickly to almost perfect detection after $k = 50$. Therefore, subsampling at small fixed values of $k$ will show improved bias, although the best improvements will be shown by selecting $k$ to extremely small values in the range [2, 10]. However, these same (bias-centered) improvements can be realized simply by running the base method once on the full data with larger values of $k$. The improved performance with $k$ can also be realized to a limited degree in related data distributions. For example, if one generated multiple Gaussian clusters and treated the pareto-extremes of the clusters as outliers, then the accuracy on the base data will increase with $k$ only for very small values of $k$. In such cases, subsampling with very small values of $k$ will show improvement on the individual component detectors, because such values of $k$ are suboptimal for the base (full) data.

The effects of the bias in subsampling can help or hurt the component detector performance in an unpredictable way. The salvation lies only in the variance reduction effects of the averaging process in the ensemble, as in classification [6; 7; 8]. However, this salvation is not guaranteed, when there is significant deterioration in base detector performance with respect to full data performance because of the unpredictability in bias characteristics.

**A Correct View of Subsampling**

It is evident from the aforementioned discussion that all ensemble methods in classification, which do not require the labels to be observed, can be trivially generalized to outlier detection. However, the lack of observed labels does cause challenges. For example, when using subagging in classification, one can optimize the parameters for the subsample with cross-validation. This is not possible in unsupervised problems like outlier detection, and the unpredictable performance of the base detectors can sometimes be a liability even after the variance reduction of the ensemble. This tends to make the overall effect of subagging more unpredictable in outlier detection, as compared to data classification. These unsupervised aspects are where outlier ensembles are truly different from classification ensembles.

Although the adaptation of subsampling from classification is a good idea, the paper [24] does not cite, relate to, or credit the existing subsampling (subagging) ideas in classification [6; 7; 8], of which this work is a direct derivative. Practically, there are very limited differences (both in theory and experimental frameworks) between subsampling for classification and for outlier detection, compared to other applications like clustering. Only those conclusions in [24], which are consistent with known ensemble theory in data classification, are correct. The portions on improvement of base detectors seem not to be true in general. In fact, the (new) assertions on the improvement of the performance of individual detectors can cause confusion in students and young researchers trying to develop new ensemble algorithms. This type of incorrect theory obfuscates what would otherwise be a simple, easily understood, and useful adaptation from classification. It also distracts one from looking for a real solution to the unpredictability of subsampling with base subsample size, which is where the problem is truly different from classification.

The main error in the theoretical results of [24] arises from use of the unnormalized $k$-NN distance gap between outliers and inliers in lieu of probability densities. One cannot make any inferences from this (unnormalized) gap increase without accounting for the corresponding increase in (unnormalized) score variance. Divergence in absolute values of scores makes no difference to the ranks in the outlier scores when all score values are scaled up by the same factor of $(1/f)^{1/d}$.

A simple example is where all scores are multiplied by 2, which results in divergence of scores between outliers and inliers but no impact on the outlier detector. This is because variances are scaled up by $2^2 = 4$. Subsampling increases the variances of the scores significantly in a single ensemble component (even after scaling adjustments) because of less training data; this increases the probability of inversion. This is the reason the experiments in Figures 1(c), (f), (i) and the experiments in Figures 2(c), (f), (i), both show accuracy reduction after (roughly) adjusting the value of $k$ for the bias effects; another way of understanding this is that less data increases the error from increased variance effects.

**Interactions between Base Detector and Ensemble**

The overall performance of subsampling will depend on the specific choice of the base detector. For example, if the base detectors output highly correlated scores, then subsampling will not help very much because of poor variance reduction. There are also some unusual cases in which subsampling can perform significantly worse than all the base detectors when measured in terms of the AUC. For example, LOF sometimes sets the scores of some points in the neighborhood (see Figure 5) of repeated (duplicate) points to be $\infty$. This is a weakness in algorithm design, especially since many of these}

---

Figure 4: The AUC-vs-$k$ always increases at small values of $k$ in the normal distribution. Therefore, subsampling at very small values of $k$ would be beneficial.
predictions tend to lie in truly dense regions with lots of repeated points. In some unusual cases (see section 5), this can cause LOF to have worse-than-random bias (in expectation), even when its ROC curves show high values of the AUC over individual detectors. This occurs when different points obtain \( \infty \) scores in different ensemble components. It is only upon averaging the scores, that one finds the ensemble to be worse than its base detectors. In other words, the AUCs of individual base detectors do not reflect the full impact of the \( \infty \) scores, whereas the AUC of the averaged score reflects the expected bias of the detector more closely. Many distance-based detectors can show poor performance when unnaturally small values of \( k \) are used on large data sets. However, such values of \( k \) might be appropriate for smaller data sets (subsamples). In other words, the optimal value of \( k \) typically increases with data size for a particular base distribution. Alternatively, one can fix \( k \) at an artificially small value and reduce subsample size to create the illusion of better performance with less data. However, these effects might not be observed at larger values of \( k \).

These implications are important because they show the unexpected interactions that might occur between a base detector and an ensemble method. For example, trying to use bagging instead of subsampling with LOF can worsen the \( \infty \) problem because of repetitions in sampled points. Feature bagging can also increase the propensity to create such duplicates in the data set. In all these cases, the performance might seem surprising at first sight, although it can usually be explained from the bias-variance perspective.

### Implications for Computational Complexity

It is claimed in [24] that one can improve over a single application of the base method on the full data set with subsampling, while also improving accuracy. This is possible only for data sets, such as those in Figure 1, in which the bias helps the component detectors, and therefore a relatively small number of trials is required. When attempting to win only by variance reduction, it is important to use as much of the training data as possible in subsamples. For data sets, like those in Figure 2, where the individual component detectors perform worse than that on the full data sets, many more trials may be required for the variance reduction effects to overcome the bias limitations and it is hard to guarantee improvement in a specific number of trials, if at all.

At the end of the process, the scores of each data point in different components are averaged to create a unified score. However, before averaging, the \( n_0 \) outlier scores from each detector should be standardized to zero mean and unit variance. This standardization is necessary because subsamples of different sizes will create outlier scores of different raw values for unnormalized KNN-algorithms. We refer to this approach as Variable Subsampling (VS). It is noteworthy that the subsampling approach always selects between 50 and 1000 data points irrespective of base data size. For data sets with less than 1000 points, the maximum raw size would be equal to the size of the data set. For data sets with less than 50 points, subsampling is not recommended. We now analyze the effect of such an approach on parameter choice, by using the \( kNN \)-algorithm as an example. The merit of this approach is that it effectively samples for different values of model parameters. For example, varying the subsample size at fixed \( k \) effectively varies the percentile value of \( k \) in the subsample. In general, holding data size-sensitive parameters fixed, while varying subsample size, has an automatic effect of parameter space exploration. If we view each component detector after selecting the subsample size, then it has a bias, which is component dependent. However, if we view the randomized process of selecting the subsample size as a part of the component detector, then every component has the same bias, and the variability in the aforementioned component-dependent bias now becomes a part of this detector variance. One can reduce this variance with ensembling, with the additional advantage that the underlying component detectors of variable subsampling tend to be far less correlated with one another as compared to

### 3.3 Variable Subsampling

The unpredictable performance of component detectors in subsampling will also be reflected in the final results from the ensemble, even after variance reduction. In such cases, it is indeed possible for the ensemble to perform worse than the base detectors. We will experimentally show several examples of this phenomenon later.

How can one address these challenges and fix subsampling to address challenges, which are specific to outlier detection, and not faced in classification? The simplest solution to this problem is to vary the subsampling rate. As we will see, varying the subsampling rate results in more diverse detectors. Let \( n_0 \) be the number of points in the base data set \( D \). The algorithm proceeds as follows:

1. Select \( f \) uniformly at random between \( \min\{1, \frac{20}{n_0}\} \) and \( \min\{1, \frac{1000}{n_0}\} \), where \( n_0 \) is the number of points in the original data set \( D \).
2. Select \( f \cdot n_0 \) randomly sampled points from the original data \( D \), and apply the base outlier detector on this sample to create an outlier detection model. Score each point in \( D \) using this model.

Figure 5: LOF can report very large or \( \infty \) scores (false positives) at small \( k \) in very dense regions.
fixed subsampling. As a result, one can now aim for better accuracy improvements in the ensemble. Therefore, this approach provides variance reduction not only over different choices of the training data, but also over different randomized choices of \( k \) (in an implicit way). In other words, the approach becomes insensitive to specific parameterizations. Although, we have focussed on the parameterization of distance-based detectors here, it is conceivable and likely that such an approach is also likely to make ensembles created with other types of base detectors robust to both parameter and data-size-sensitive design choices. This makes the VS approach more general and desirable than simply varying the value of \( k \) across detectors; it is independent of the nature of the parameters/design choices in the base detector and it concurrently achieves other forms of variance reduction in an implicit way. For data size-sensitive parameters, it is advisable to select them while keeping in mind that subsample sizes vary between 50 and 1000 points. Knowledge of subsample sizes eases the parameter selection process to some extent. For example, for distance-based detectors, we recommend that a value of \( k = 5 \) will result in a percentile value of \( k \) varying between 0.5% to 10% of data size, which seems reasonable.

It is noteworthy that variable subsampling works with raw subsample sizes between 50 and 1000, irrespective of base data size. By fixing the subsample size in a constant range, it would seem at first sight that the approach cannot take advantage of the larger base data sizes. This is, however, not the case: larger data sets would result in less overlap across different subsamples, and therefore less correlation across detectors. This would lead to better variance reduction. The idea is to leverage the larger base data size for better de-correlation across detectors rather than build more robust base detectors with larger subsamples; the former is a more efficient form of variance reduction. After all, the number of points required to accurately model a distribution depends on the absolute subsample size, rather than on the size of the original data set obtained by the data collector. Even if we work under the implicit assumption that a data collector would collect more data for a more complex data distribution, it is unlikely that the required number of data points to accurately model the distribution varies linearly with the collected data size. If desired, one can use other heuristics to increase the robustness of base detector with increasing data size, such as selecting \( f \) from \( (\min\{1, \frac{50}{n_0}\}, \min\{1, \sqrt{\frac{1000}{n_0}}\}) \).

The maximum subsampling rate should always reduce with base data size, to increase the de-correlation benefits rather than using it only to improve the base detector.

### 3.3.1 Computational Complexity of VS

By focusing on an absolute size of the subsample, rather than a subsampling rate, we have ensured that each detector requires time linear in the base data size, rather than quadratic. This is because points in the full data set need to be scored against a subsample of constant size. Therefore, the relative speed-up increases with increasing data size. In Figure 6, we have analytically shown the number of operations of a quadratic base detector, and two variations of the subsampling approach with 100 trials. One is based on a constant maximum subsample size of 1000, and the other is based on a maximum subsample size of \( \sqrt{1000n_0} \). We assume that the base detector requires \( O(n_0^2) \) operations and a 100-trial subsampling approach requires \( 100 \times n_{\text{max}} \times 50 \times n_0 \), where \( n_{\text{max}} \) is maximum subsample size in a particular type of variable subsampling. For any data set with more than 50000 points, variable subsampling with constant subsample size has a clear advantage over a single application of the base detector, and it would be 20 times faster for a million point data set, although the figure only shows results up to 200,000 points. If one were to extend the X-axis to beyond 5 million points, even the approach using a maximum subsample size of \( \sqrt{1000n_0} \) would overtake the base detector. For larger data sizes, most of the base data points might not even be included within one of the 100 subsamples; nevertheless, the accuracy could be superior to that of a model on the (full) base data because increasing data size on a single detector is an inefficient way of reducing variance as compared to variable subsampling. The only way of consistently doing better with less data is to use a better designed technique rather than using an identical method on less data.

### 3.4 A Review of Feature Bagging

The feature bagging method [14] samples different subsets of dimensions. The basic idea is to sample a number \( r \) between \([d/2]\) and \( d - 1\), and then select \( r \) dimensions randomly from the data set. The base detector is applied to this lower-dimensional projection. The scores across various components are then averaged, although other combination methods were also proposed in [14].

Feature bagging (with averaging) is a method that reduces detector variance. Feature bagging with a particular subset of dimensions has a bias that depends on the selected dimensions. However, if one views the step of randomly selecting the subset of dimensions as a part of the component detector, then such a (randomized) detector has exactly the same bias, and the aforementioned variability in the bias across different dimension-specific instantiations now becomes a part of this (randomized) detector variance. In such cases, using an average combination is able to achieve variance reduction. The smaller the subset of dimensions selected, the greater the variance reduction. This is because the underlying detectors tend to be relatively uncorrelated if few overlapping dimensions are selected by different detectors. However, if all dimensions are informative, the bias characteristics of such an approach are likely to work against feature bagging because down-selecting the dimensions will lose information.

In this context, it needs to be pointed out that the method in [14] proposes to always randomly select between \([d/2]\)
and \( d - 1 \) dimensions; one doesn’t always gain the best variance reduction by selecting so many dimensions because of correlations between different detectors. Correlations between detectors hinder variance reduction. One might even select the same subset of dimensions repeatedly, while providing drastically worse bias characteristics. In particular, consider a 6-dimensional data set. The number of possible 3-dimensional projections is 20, the number of possible 4-dimensional projections is 15, and the number of 5-dimensional projections is 6. The total number of possibilities is 41. Therefore, most of the projections (and especially the 4 and 5-dimensional ones) will be repeated multiple times in a set of 100 trials, and not much variance can be reduced from such repetitions. On the other hand, the 3-dimensional projections, while more diverse in overlap and repetition, will have deteriorated bias characteristics. This will also be reflected in the final ensemble performance. Here, it is important to note that the most diverse dimensions provide the worst bias characteristics and vice versa. How can one improve both simultaneously?

### 3.5 Rotated Bagging (RB)

A natural solution is to devise a randomized sampling scheme that reduces the correlations among detectors. We propose to use rotated bagging, in which the data is rotated to a random axis system before selecting the features. The random rotation provides further diversity. A salient observation is that real data sets often have significant correlations, and the projections along different directions are correlated with one another. This means that we can afford to use a much lower dimensionality than \( d/2 \) to represent the data without losing too much information. In real data sets, the implicit dimensionality usually does not grow much faster than \( \sqrt{d} \) with increasing dimensionality \( d \). Therefore, we propose to use \( 2 + \lceil \sqrt{d}/2 \rceil \) orthogonal directions from the rotated axis-system as the set of relevant feature bags. Using a lower dimensional projection helps in increasing diversity and therefore it leads to better variance reduction. At the same time the \( 2 + \lceil \sqrt{d}/2 \rceil \) dimensions are able to roughly capture most of the salient modeling information in the data because of the random orientation of the axis system. In other words, one is able to increase the potential of better variance reduction without compromising bias too much. The approach is not designed to work for 3 or less dimensions. Therefore, a constant value of 2 is added up front to prevent its use in such cases. The component detectors will be more uncorrelated in high dimensional cases, which yields a better opportunity for variance reduction. The overall algorithm works as follows:

1. Determine a randomly rotated axis system in the data.
2. Sample \( r = 2 + \lceil \sqrt{d}/2 \rceil \) directions from rotated axis system. Project data along these \( r \) directions.
3. Run the outlier detector on projected data.

After running the detector, the scores can be averaged with a primary goal of variance reduction. It is important to use standardization on the scores before the combination. However, other choices for combination are possible, which will be discussed in a later section.

How can one determine \( r = 2 + \lceil \sqrt{d}/2 \rceil \) randomly rotated mutually orthogonal directions? The basic idea is to generate a \( d \times r \) random matrix \( Y \), such that each value in the matrix is uniformly distributed in \([-1,1]\). Let the \( t \)-th column of \( Y \) be denoted by \( \overline{y}_t \). Then, the \( r \) random orthogonal directions \( \overline{y}_1 \ldots \overline{y}_r \) are generated using a straightforward Gram-Schmidt orthogonalization of \( \overline{y}_1 \ldots \overline{y}_r \) as follows:

1. \( t = 1; \overline{y}_t = \overline{y}_1 \)
2. \( \overline{y}_{t+1} = \overline{y}_{t+1} - \sum_{j=1}^{t} (\overline{y}_{t+1} \cdot \overline{y}_j) \overline{y}_j \)
3. Normalize \( \overline{y}_{t+1} \) to unit norm.
4. \( t = t + 1 \)
5. if \( t < r \) go to step 2

Let the resulting \( d \times r \) matrix with columns \( \overline{y}_1 \ldots \overline{y}_r \) be denoted by \( E \). The \( n_0 \times d \) data set \( D \) is transformed and projected to these orthogonal directions by computing the matrix product \( DE \), which is an \( n_0 \times r \) matrix of \( r \)-dimensional points. We refer to this approach as Rotated Bagging (RB).

### 3.6 Variable Subsampling with Rotated Bagging (VR)

It is possible to combine the base detectors in variable subsampling and rotated bagging to create an even more diverse base detector. This will help in variance reduction. Furthermore, because of the reduction in terms of both points and dimensions, significant computational savings are achieved. The combined base detector is created is as follows:

1. Project the data into a random \( 2 + \lceil \sqrt{d}/2 \rceil \)-dimensional space using the rotation method of the previous section.
2. Select a variable size subsample using the approach described in section 3.3.
3. Score each point using the reduced data set.

The scores of these individual detectors can then be combined into the final ensemble score. It is important to use Z-score normalization of the scores from the base detectors before combination. We refer to this approach as variable subsampling with rotated bagging (VR).

### 3.7 Observations on Computational Benefits

Rotated bagging has clear computational benefits because one is using only \( \sqrt{d} \) dimensions. With increasing dimensionality the benefit increases. When combined with variable subsampling, the benefits can be very significant. For example, for a data set containing ten million points and 100 dimensions (i.e., a billion entries), each ensemble component would use a data matrix of size at most \( 1000 \times 7 \) (i.e., less than a ten-thousand entries). In space-constrained settings, this can make a difference in terms of being able to use the approach at all. For 100 trials, the ensemble (containing quadratic base detectors) would be hundreds of times faster than a single application of the base method on the full data.

### 3.8 Other Variance Reduction Methods

As discussed earlier, the similarity of the bias-variance trade-off in outlier detection to that of classification means that one can trivially adapt many classification ensemble algorithms to outlier detection. For example, bagging, bragging, wagging, subagging, and various forms of diversity incorporation can be easily adapted to outlier detection. With some
methods such as bootstrapped aggregation, care should be taken to use detectors that perform robustly in the presence of repeated instances. Using LOF as a base detector would be a bad idea, without proper handing of repeated instances within the implementation. There is even a rich literature on diversity incorporation by artificially adding training data or otherwise perturbing the training data [16]. Note that the work in [25] is a variation of this basic idea, although it provides a different theoretical justification. New theoretical arguments do not need to be invented for the effectiveness of these methods, because they follow trivially from the arguments used in classification. Even the benchmarking of these outlier detection ensembles is done within a supervised framework.

### 3.9 Ideas for Bias Reduction

Bias reduction is, however, a completely different matter. In outlier detection, it is very hard to reduce bias in a controlled way, although some heuristic ideas are possible based on common observations about “typical” data and how the outlier scores might behave in typical data. Even then, there is no guarantee that such heuristic methods will always reduce bias. The main problem with attempting bias reduction is that most such methods in classification use knowledge of the labels in intermediate steps. This is not possible in unsupervised problems like outlier detection.

An example of a bias reduction approach, which is used commonly in classification, is boosting [10]. Boosting uses the labels for evaluation in the intermediate steps of the algorithm. This requirement rules out its adaptation to outlier detection. It has been suggested [23] that one might be able to substitute internal validity measures for the ground truth in methods like boosting. However, the problem with such an approach is that internal validity measures have built-in biases of their own and the results can be misleading.

Trying to use an internal validity measure for boosting is a circular argument because all validity measures need to use a model that will have a built-in bias; the bias reduction of the boosted algorithm would then be at the mercy of the quirks (i.e., bias) of this internal validity model. In general, internal validity measures are not fully trusted even in clustering where they have specific biases in favor of particular algorithms. In the context of outlier detection, the problem is even more significant because a small number of errors in evaluating outlier points can have drastic results.

One commonly used heuristic approach, which is discussed in [2], is to remove outliers in successive iterations in order to build a successively more robust outlier model iteratively. This is a type of sequential ensemble. The basic idea is that outliers interfere with the creation of a model of normal data, and the removal of points with high outlier scores will be beneficial for the model in the next iteration. Although it is not guaranteed that the correct data points might be removed, the advantages outweigh the risks, and the approach has indeed been used successfully in the past [5] in an indirect (non-ensemble) form. A softer version of this approach is to simply down-weight points with high outlier scores in subsequent iterations to ensure that outlier points do not overly influence the normal model of points. One can implement this type of down-weighting with biased sampling; this has the additional benefit of reducing variance. Of course, such an approach is not exactly the same as how boosting is understood in the classification literature, where one combines the knowledge from multiple components in a more holistic way. Nevertheless, it has the same overall effect of bias reduction.

### 4. OUTLIER SCORE COMBINATION

Given the outlier scores from various detectors, a final step of ensemble-based approach is to combine the scores from various detectors. Let us consider the case of a set of $m$ independent detectors, which output the scores $s_1(i) \ldots s_m(i)$ for the $i$th data points. When the scores are produced by detectors of different types, it is assumed that they are standardized. There are two commonly used combination functions:

1. **Averaging:** The average of the scores $s_1(i) \ldots s_m(i)$ is reported as the final score of the $i$th data point.

2. **Maximum:** The maximum of $s_1(i) \ldots s_m(i)$ is reported as the outlier score.

Which of these methods of model combination is better? It has been suggested [23] that the averaging variant is better and that the maximum function overestimates the absolute scores [23] by picking out the larger errors. On the other hand, the work in [14] shows some comparative experimental results between the averaging function and a rank-based variant of the maximization function (referred to as breadth-first combination in [14]). The results are data-dependent and do not seem to show clear superiority of one method over the other.

A clearer picture may be obtained from the bias-variance trade-off. The effect of averaging is very clear because it results in a reduction of the variance (as in classification). We argue, however, that the specific choice of the combination function often depends on the data set at hand. In real settings, one is often able to de-emphasize irrelevant or weak ensemble (poorly biased) components with the maximization function. Therefore, one is often able to reduce bias. However, the maximization function might increase variance, especially for small training data sets. The specific effect will depend on the data set at hand, which is also reflected in the results of [14]. This is yet another example of the power of the venerable bias-variance trade-off in understanding all types of ensemble analysis. In our experiments, we found that it was (mostly) in smaller data sets and subsample sizes (i.e., where variance was large), that averaging performed better than maximum.

Next, we explain the bias reduction effects of the maximization combination. In many “difficult” data sets, the outliers may be well hidden, as a result of which many ensemble components may give them inlier-like scores. In such cases, the scores of outlier points are often relatively underestimated in most ensemble components as compared to inlier data points. In order to explain this point, let us consider the feature bagging approach of [14], in which the outliers are hidden in small subsets of dimensions. In such cases, depending on the nature of the underlying data set, a large majority of subspace samples may not contain many of the relevant dimensions. Therefore, most of the subspace samples will provide significant underestimates of the outlier scores for the (small number of) true outlier points and mild overestimates of the outlier scores for the (many) normal points. This is a problem of bias, which is caused by the well hidden nature of outliers. We argue that such kinds of
bias are inherent\(^9\) to the problem of outlier detection. The scores of outlier points are often far more brittle to small algorithm modifications, as compared to the scores of inlier points. Using a maximization ensemble is simply a way of trying to identify components in which the outlier-like behavior is best magnified. Of course, it is fully understood that any bias-reduction method in an unsupervised problem like outlier detection is inherently heuristic, and it might not work for a specific data set. For example, if a training data set (or subsample) is very small, then the maximization function will not work very well because of its propensity of picking out the high variance in the scores. Clearly, there are trade-offs between the use of the maximization and averaging function and it is difficult to declare one of them as a clear winner. This point also seems to be underscored by the experimental results presented in [14], where the relative behavior of the two methods (i.e., averaging versus maximum rank) depends on the specific data set. In this paper, we will provide experimental results which show further insights.

### 4.1 A Simple Example

In order to illustrate this point, we will provide a simple example of a toy data set \( T \) and ensemble scheme, where outliers are well hidden in the data set. Consider the case, where a data set has exactly \( n \) data points and \( d \) dimensions. For the purpose of discussion, we will assume that the value of \( d \) is very large (e.g., a few hundred thousand). Assumes that the data set contains a single outlier. For the \((n-1)\) normal data points, the data is uniformly distributed in [-1, 1]. The distribution for the outlier point is slightly different in that a randomly chosen dimension has a different distribution. On exactly \((d-1)\) dimensions, the outlier point is again uniformly distributed in [-1, 1]. On the remaining (randomly chosen) dimension, the value of the corresponding attribute is in the range [2, 3].

Note that the single outlier can be trivially discovered by many simple heuristics, although many off-the-shelf distance-based algorithms might not do very well because of the averaging effects of the irrelevant dimensions. In practice, an outlier detection algorithm is not optimized to any particular data set, and one often uses detectors which are not optimized to the data set at hand. For example, consider the case where the base detector is an extreme value analysis method [22] in which the distance from the data mean is reported as the outlier score. Note that the data distribution of \( T \) is such that the mean of the data can be approximated to be the origin in this case. The ensemble method is assumed to be a variant of the feature bagging scheme [14], in which each dimension in the data is selected exactly once and the detector is applied on this 1-dimensional data set. The process is repeated for each of the \( d \) dimensions, and the final score can be reported using either the averaging or the maximum function over these \( d \) different scores. Therefore, our simple ensemble-based approach has \( d \) components. We will derive the probability that the score for an outlier point is greater than that for an inlier point under both the averaging and maximization schemes. In other words, we would like to compute the probability of a rank inversion in the two cases.

\(^9\)The original LOF paper recognized the problem of dilution from irrelevant ensemble components and therefore suggested the use of the maximization function.

The averaging function will yield a combination score for the inlier points, which has an expected value of 0.5 because each score is randomly distributed in (0, 1). The variance of the score is \( 1/(12 \cdot d) \) over the different ensemble components. On the other hand, by using the same argument, the outlier point will have an expected score of \( 0.5(d-1)+2.5/d \), because the irrelevant dimensions contribute \( 0.5(d-1)/d \) to the expected value, and the single relevant dimension contributes \( 2.5/d \) to the expected score. The variance of the score is \( 1/(12d) \). Therefore, the difference \( M \) between the two scores will be a random variable with an expected mean of \( \mu = 2/d \) and a variance of \( \sigma^2 = 1/(6d) \). Furthermore, this random variable \( M \) will be normally distributed when \( d \) becomes large. Note that an inversion between the outlier and a randomly selected inlier occurs when \( M \) is negative. Let \( Z \sim N(0, 1) \) be a random variable drawn from the standard normal distribution with 0 mean and unit variance. Therefore, we have:

\[
P(\text{Inversion}) = P(M < 0) = P(Z < 0 - \mu/\sigma) = P(Z < -2\sqrt{6/d})
\]

Note that the expression \( 2\sqrt{6/d} \) tends to zero with increasing dimensionality, and the resulting probability evaluates to almost 0.5. This means that with increasing dimensionality, an inlier is almost equally likely to have a larger outlier score than a truly outlier point. In other words, the averaging approach increasingly provides performance that is similar to a random outlier score for each point. This is because the data point becomes increasingly hidden by the irrelevant dimensions, and the averaging function continues to dilute the outlier score with increasing dimensionality.

Nevertheless, the maximization function always discovers the correct relative score of the outlier point with respect to the inlier points because it always reports a value in the range [2, 3], which is greater than the outlier score of the other data points. In other words, the maximization ensemble properly corrects for the natural bias of outlier detection algorithms, in which the scores of well-hidden outliers are often more unstable than inliers. In the easy cases, where most outliers are “obvious” and can be discovered by the majority of the ensemble components, the averaging approach will almost always do better by reducing variance effects. However, if it can be argued that the discovery of “obvious” outliers is not quite as interesting from an analytical perspective, the maximization function will have a clear advantage.

### 4.2 Using Ranks

A related question is whether using ranks as base detector output might be a better choice than using absolute outlier scores. After all, the metrics for outlier detection are based on the rank-wise AUCs rather than the score-wise MSEs. Ranks are especially robust to the instability of raw scores of the underlying detectors. For example, the \( \infty \)-problem of LOF would affect the absolute scores but it would affect the ranks to a much smaller degree. However, ranks do lose a lot of relevant information, when scores convey the true degree of outlierness. In such cases, using ranks could increase bias-centric errors, which might also be manifested in the ranks of the final combination score. Therefore, while ranks might work well with some base detectors, their improved behavior is certainly not universal.
4.3 Balanced Choices

Clearly, the bias-variance trade-off suggests that different combination functions might do better in different settings. Therefore, it is natural to balance the effort in reducing bias and variance by combining the merits of the two methods. We propose two schemes, both of which normalize to Z-scores before applying the combination:

**AOM Method:** For \( m \) ensemble components, we divide the components into approximately \( m/q \) buckets of \( q \) components each. First, a maximization is used over each of the buckets of \( q \) components, and then the scores are averaged over the \( m/q \) buckets. Note that one does not need to assign equal resources to maximization and averaging; in fact, the value of \( q \) should be selected to be less than \( m/q \). For our implementations, we used 100 trials, with \( q = 5 \). We refer to this method as AOM, which stands for Average of Maximum.

**Thresh Method:** A method suggested in [2], for combining the scores of multiple detectors, is to use an absolute threshold \( t \) on the (standardized) outlier score, and then adding the (thresholded and standardized) outlier scores for these components. The threshold is chosen in a mild way, such as a value of \( t = 0 \) on the standardized score. Note that values less than 0 almost always correspond to strong inliers. The overall effect of this approach is to reward points for showing up as outliers in a given component, but not to penalize them too much for showing up as strong inliers. For our implementations, we always used a threshold value of \( t = 0 \) on the Z-score. An important point is that such an approach can sometimes lead to tied scores among the lowest ranked (i.e., least outlier-like) points having a score of exactly \( m \cdot t \). Such ties are broken among the lowest ranked points by using their average standardized score across the \( m \) ensemble components. As a practical matter, one can add a small amount \( \epsilon \approx \text{avg} \), proportional to the average standardized score \( \text{avg} \), of such points, to achieve the desired tie-breaking. We refer to this approach as Thresh.

The AOM combination scheme is particularly useful when the maximum number of trials is not a concern from the computationally efficiency perspective. For example, with averaging, we found that it was often hard to do much better by significantly increasing the number of trials beyond a certain point. However, to saturate the benefits of combining maximization and averaging (e.g., AOM) one would need a larger number of trials. Nevertheless, in this paper, we show that even with the same number of trials, schemes such as AOM perform quite well. With faster base detectors, one can run a far larger number of trials to gain the maximum accuracy improvements from both bias and variance reduction; indeed many of the ensemble methods proposed in this paper also provide the dual benefit of greater speed. The Thresh method can be viewed as a faster way of combining bias and variance reduction, when computational efficiency is important. Other ideas for combining bias and variance reduction include the use of Maximum-of-Average (MOA).

5. EXPERIMENTAL RESULTS

In this section, we provide experimental results showing the relative effectiveness of various ensemble methods. We used the average \( k \)-NN and LOF methods as base detectors.

5.1 Data Set Descriptions

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Points</th>
<th>Attributes</th>
<th>Percentage Outliers (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>214</td>
<td>9</td>
<td>4.2</td>
</tr>
<tr>
<td>Lymphography</td>
<td>148</td>
<td>18</td>
<td>4.1</td>
</tr>
<tr>
<td>WBC</td>
<td>378</td>
<td>30</td>
<td>5.6</td>
</tr>
<tr>
<td>Vowels</td>
<td>1456</td>
<td>12</td>
<td>3.4</td>
</tr>
<tr>
<td>Thyroid</td>
<td>3772</td>
<td>6</td>
<td>2.5</td>
</tr>
<tr>
<td>Satimage-2</td>
<td>5803</td>
<td>36</td>
<td>1.2</td>
</tr>
<tr>
<td>Cardio</td>
<td>1831</td>
<td>21</td>
<td>9.6</td>
</tr>
<tr>
<td>Optdigits</td>
<td>5126</td>
<td>64</td>
<td>2.9</td>
</tr>
<tr>
<td>Musk</td>
<td>3062</td>
<td>166</td>
<td>3.2</td>
</tr>
</tbody>
</table>

We used nine data sets from the UCI Machine learning repository\(^{10}\). In some cases, further preprocessing was required. In cases where one of the classes was already rare, it was labeled as the outlier class. In cases where a data set contained relatively balanced classes, downsampling was necessary to create an outlier class. In some cases, multiple large classes were combined to create inliers and multiple minority classes were combined to create outliers. In the following, we provide a brief description of the data preparation process.

The Glass data set contained attributes regarding several glass types. Here, points of class 6 were marked as outliers, while all other points were inliers. For the Lymphography data set classes 1 and 4 were outliers while the other classes were inliers. The Wisconsin-Breast Cancer (Diagnostics) data set (WBC) contained malignant and benign classes, and we started with a processed version\(^{11}\) of the data set. We further downsampled the malignant class to 21 outliers, while points in the benign class were considered inliers. In the Japanese Vowels (Vowels) data set, we treat each frame in the training data as an individual data point, whereas the UCI repository treats a block of frames (utterance) as an individual point. In this case, class (speaker) 1 was downsampled to 50 outliers. The inliers contained classes 6, 7 and 8. Other classes were discarded. The ANN-Thyroid data set is the same as that in [13]. In the Statlog (Landsat Satellite) data set, the training and test data were combined. Class 2 was downsampled to 71 outliers, while all the other classes were combined to form an inlier class. Our modified data set is referred to as Satimage-2. The Cardiotocography (Cardio) data set contained measurements taken from foetal heart rate signals. The classes in the data set were normal, suspect, and pathologic. The normal class formed the inliers, while the pathologic (outlier) class was down-sampled to 176 points. The suspect class was discarded. In Optdigits, instances of digits 1-9 where inliers and instances of digit 0 were down-sampled to 150 outliers. The Musk data set contained several musk and non-musk classes. We combined non-musk classes j146, j147, and 252 to form the inliers, while the musk classes 213 and 211 were added as outliers without down-sampling. Other classes were discarded. Refer to Table 1 for details of data sets.

5.2 Ensemble Combination Methods

In each case, 100 trials of the base detector were used. The base methods are combined in four different ways.

1. Averaging: This is the averaging combination method,

\(^{10}\)http://archive.ics.uci.edu/ml/datasets.html

\(^{11}\)http://www.ipd.kit.edu/~muellere/HiCS/
in which the scores from different base detectors are averaged. In the case of the $k$-NN detector, the scores are also normalized to $Z$-values before averaging. All the three new schemes, corresponding to (variable subsampling (VS), rotated bagging (RB), and variable subsampling with rotated bagging (VR)), are always normalized of $Z$-values before averaging, because of the large variations in the scores produced by these methods. The results for averaging are shown as a triangle in each box plot of Figures 7, 8, and 9.

2. **Maximization**: All scores from all algorithms are first converted to $Z$-values. Then, the maximum scores across all ensemble components for each data point are reported. The maximization ensemble scores are shown with an ‘x’ in Figures 7, 8, and 9.

3. **Average-of-Maximum (AOM)**: The 100 trials were divided into 20 buckets of size 5 each. The maximum $Z$-score was taken over each bucket of size 5. Then, the resulting 20 scores for each point were averaged. The ensemble performance is shown with a circle in Figures 7, 8, and 9.

4. **Threshold sum (Thresh)**: All non-negative $Z$-scores for each data point were added up over the 100 components to create the unified score. Tie-breaking of lowest-ranked points is performed as discussed earlier. The ensemble performance is shown with a square in Figures 7, 8, and 9.

5.3 **Normalization of Base Detectors**
The outlier scores in an average $k$-NN detector are not comparable in different ensemble components, especially when using methods like feature bagging and variable subsampling. Therefore, all ensemble scores using the average $k$-NN detectors were normalized to $Z$-values. For $LOF$, which already produces normalized scores, the scores were not re-normalized to $Z$-scores for the averaging ensemble in the case of fixed subsampling and feature bagging. This was done in order to be consistent with their original implementations. However, for the maximization and balanced methods, we always re-normalized to $Z$-scores across all ensemble and base methods (including $LOF$). Furthermore, for the new methods proposed (Variable Subsampling, Rotated Bagging, and the combination), we always re-normalized, irrespective of the nature of the combination method used. This is because these detectors often contain components with such widely varying bias characteristics, that re-normalization was essential to make them comparable. An important quirk in the case of $LOF$ was the case when at least one outlier score was $\infty$. In such cases, the $\infty$-scores were excluded while computing the mean and standard deviation for normalization.

5.4 **Performance Results**
We performed the tests over the nine data sets discussed earlier. We tested using both the average KNN-detector and the $LOF$ detector at values of $k = 5$ and $k = 10$. Two different values of $k$ were used because the performance results (and even the trends with varying subsampling rates) were found to be sensitive to the values of $k$. In each case, we show the following 14 ensemble methods:

1. **Fixed subsampling**: This is the approach used in [24] at varying subsampling rates starting from 0.1 to 1.0. Note that this results in a total of 10 box plots. The box-plot at 1.0 corresponds to the base detector.

2. **Variable subsampling (VS)**: This approach always samples between 50 and 1000 points from the data set. When the data set contained less than 1000 points, the upper bound was set to the data size. Note that this type of variable subsampling explores components with different bias characteristics within the ensemble. This scheme is annotated as VS in the figures.

3. **Feature bagging**: This is the feature bagging method discussed in [14]. This scheme is annotated as FB in the figures.

4. **Rotated Bagging**: This is the rotated bagging scheme discussed in the paper, which is annotated as RB.

5. **Variable Subsampling with Rotated Bagging (VR)**: This is the combination of the variable subsampling approach with rotated bagging. The scheme is annotated as VR in the figures.

The performances of each of the methods are discussed in Figures 7, 8, and 9, respectively. For each data set, there are four figures corresponding to the two detectors, and values of $k$ set to 5 and 10, respectively. The box-plots in each figure are shown at varying levels of fixed subsampling rates, feature-bagging, and other methods introduced in this paper. Here, we summarize the key findings of our method:

1. Contrary to the claims in [24], smaller subsamples do not always lead to superior performance for the base detectors. In particular, the trends depend on the AUC-vs-$k$ curves as discussed earlier, and also on the selected value of $k$. If the selected value of $k$ is sub-optimally small, then it is possible for subsampling to improve the base detector performance. Note that we used reasonably small values of $k$ in our experiments ($k = 5$ and $k = 10$), and yet, the subsampling did not always improve the base detector performance. In fact, for the case of the unnormalized average $k$-NN detector, significant inversion in base detector performance was observed for only 3 of the 9 data sets. Furthermore, the trends are sometimes different between $LOF$ and average $k$-NN, and also between $k = 5$ and $k = 10$. This makes the trends unpredictable but they can be fully explained by the AUC-vs-$k$ trends. The inversion was observed more frequently in the larger data sets because the values of $k$ set to 5 and 10 are suboptimally small for such cases. Furthermore, a better ensemble lift was obtained with smaller subsample sizes (but not in base detectors) because of better variance reduction.

2. In cases, where smaller subsamples showed poor base detector performance, the effects on the ensemble performance were quite significant. In many cases, the better variance reduction of smaller subsamples is not able to overcome the poor base detector performance. This is particularly true for the average $k$-NN detector algorithm. However, variable subsampling could often perform robustly irrespective of the AUC-vs-$k$
Figure 7: Performance of all ensemble methods (Glass, Lymphography, and WBC)
Figure 8: Performance of all ensemble methods (Vowels, Thyroid, Satimage-2)
Figure 9: Performance of all ensemble methods (Cardio, Optdigits, and Musk)
trends, choice of $k$, and the base algorithm. This is because variable subsampling was able to effectively ensemble over different percentile values of $k$ by varying subsample size. Therefore, irrespective of the effect of subsample size on the bias, the variable subsampling approach was able to perform effectively. More importantly, variable subsampling reduces the unpredictability in performance for a fixed subsample size.

3. We (surprisingly) found that the average $k$-NN detector usually performed better than LOF both on the base detector and the final ensemble performance (see section 5.6 for more details). The average $k$-NN detector was also relatively stable to the selection of various subsample cohorts, and this is reflected by the “thin” boxplots of these methods where the upper and lower ends of the boxes are close together. The thin box plots occur frequently in the fixed subsampling method, where every ensemble component has similar bias. The LOF method showed more variation across different executions of the base detector. This was primarily due to the instability of harmonic normalization. This instability can be viewed as a type of diversity that helps variance reduction and results in better incremental improvement of LOF over base detectors. However, the instability of LOF also led to poorer (bias-centric) performance of the base detectors; as a result the overall ensemble performance of LOF is poorer. The thin box plots of the average $k$-NN detector (for fixed subsampling) also meant that one could not obtain much variance reduction advantages from subsampling in the case of the superior (average $k$-NN) detector. This type of bias-variance trade-off is common in ensemble settings, where one must design the ensemble components to extract the maximum advantages.

4. Since variable subsampling showed more variance across different base components, the box-plots are thicker even in the case of the $k$-NN detector, and a greater advantage of subsampling was obtained. However, variable subsampling might sometimes have poorer median base detector accuracy compared to the best fixed-subsampling rate. The final ensemble performance of VS was often competitive to or better than the best rate of fixed subsampling (which varied across data sets). Note that it is impossible to know the optimal (fixed) subsampling rate for a particular data set a priori. Variable sampling solves this dilemma and thereby reduces the unpredictability of the approach.

5. Rotated bagging often performed better than feature bagging. In most cases, rotated bagging did not perform as well as variable subsampling. The real advantage of rotated bagging was obtained by combining it with subsampling.

6. When rotated bagging was combined with variable subsampling, the performance was improved slightly over many larger/high dimensional data sets. More importantly, since the combination approach reduces the data set size both in terms of the number of points and the number of dimensions, the approach is extremely fast. Therefore, the primary advantage of the combination approach is one of efficiency. These efficiency advantages can also be made to translate to better accuracy if needed. Like all other ensemble methods, we used only 100 trials for VR. Because of the greater computational efficiency of VR, it is possible to use many more trials (at the same computational cost) to achieve even better accuracy.

7. Although the averaging ensemble performed quite well, it was certainly not the best combination method over all data sets. In fact, the maximization ensemble performed better than averaging in most of the larger data sets. However, it often performed rather poorly in smaller data sets (or subsample sizes) because it fails to reduce variance. This suggests that the maximization ensemble should not be used for smaller data sets (and subsample sizes) where it can pick out the unstable noise in the scores. However, both the balanced choices (which combine bias and variance reduction), almost always performed better than averaging. Furthermore, we used 100 trials for all combination methods; this often saturates variance reduction for averaging but not for methods like AOM, where further gains are possible (by increasing the averaging component) when computational time is not an issue.

8. Feature bagging (with LOF) sometimes performed worse than applying LOF on the base data with all dimensions included. This was, in part, because of the loss of information associated with dropping dimensions. However, this cannot fully explain the performance in some data sets like Lymphography. In Lymphography, the box plots of the component LOF detectors in feature bagging were excellent (see Figure 7(e)) with (most) AUCs above 0.8; yet, the ensemble provided near-random performance. Note that the average $k$-NN detector does not show this behavior, and the peculiar performance is restricted to LOF. Furthermore, we found the ensemble performance to vary significantly across runs in such cases. What explains this unusual behavior?

This is a case where dropping dimensions leads to repeated instances in the data set. As a result, some (inlier) points have $\infty$ scores for LOF. When the scores are averaged across many components, a very large fraction of the inliers end up with $\infty$ scores. Because of the $\infty$ scores, the bias performance of the base detectors are unusually poor, but it is realized only in the AUC of the ensemble, rather than the AUC of the base detectors. This is because most base detectors contain only a small number of $\infty$ values (or a small number of base detectors contain most of the $\infty$ values). Therefore, the expected scores of many data points are $\infty$ over many runs in spite of the high AUCs. By increasing the number of trials to 1000, virtually all data points get $\infty$ scores. This example also illustrates that the variance reduction of averaging is optimized towards metrics like MSE (as in classification), which may not always be translated to correctness in ranks. Therefore, the rank-centric AUCs can occasionally perform worse than all the base detectors in some unusual settings. In some cases, rank-centric detector outputs can be effective for ensembling [14], although the behavior is not universal across detectors.
or data sets. The unusual behavior in Lymphography occurs at \( k = 5 \) rather than at \( k = 10 \), although some runs at \( k = 10 \) also deteriorated. This is because harmonic normalization is more unstable at small values of \( k \), where small groups of repeated points (or tight clusters) can throw off the computation. This is also a cautionary tale for attempting to use LOF with methods like bagging, which create repeated points in the data via bootstrapping. Although it is possible to use bagging for variance reduction in outlier detection, care must be taken to use base detectors, which are not sensitive to the presence of repeated points.

### 5.5 Recommendations for Score Combination

The aforementioned experiments suggest a number of general principles for score combination from methods:

1. The averaging method is a low risk-low reward scheme, as it always reduces variance. The performance improves over the base detectors most of the time, although significant improvements are usually not observed. It is particularly desirable for smaller data sets, because of its robustness.

2. The maximization method is a high-risk-high-reward scheme, which provides (heuristic) bias-centric improvements in many cases, but it can sometimes also increase variance. Therefore, it occasionally deteriorates below the base detector, especially in smaller data sets and subsample sizes, where it is contraindicated. The maximization function often emphasizes different outliers than the averaging function, which are well-hidden. Often, an analyst may be interested at looking at a different set of results to obtain a different perspective.

3. The balanced schemes provide a reasonably modest reward, at low risk. The gains over averaging were significant enough in so many cases, that these methods could be considered more desirable techniques than pure averaging. These schemes gain their power from their ability to combine the bias reduction in the maximization scheme with variance reduction to significantly lower the risk profile of the maximization detector, while retaining most of the performance gains.

### 5.6 Impact of Base Detectors

An important observation in Figures 7, 8, and 9, is that the LOF method generally gains greater advantage from the ensembling method as compared to the average \( K \)-NN method. This is not particularly surprising; the harmonic mean normalization is somewhat unstable, and therefore LOF has a better scope for improvement as compared to the average \( k \)-NN methods. However, how does the final ensemble performance of LOF compare to the average \( k \)-NN detector? It is here that we found some surprising results.

It is generally an article of faith in the research community that LOF is a superior detector compared to unnormalized \( k \)-NN methods. It is indeed true that LOF generally performs better than an exact \( k \)-nearest neighbor detector, in which the distance to the exact \( k \)-nearest neighbor is used as the outlier score. Most existing comparisons between LOF and unnormalized distances are based on the exact \( k \)-nearest neighbor, and the performance of the average \( k \)-NN detector has rarely been compared comprehensively to LOF. A surprising result was that we found the average \( k \)-NN detector to be superior even to LOF on the vast majority of data sets we tested.

Subsampling is a useful tool for comparing two detectors. By fixing \( k \) and varying the subsampling rate, one can effectively test varying bias settings on the data set because the percentile value of \( k \) varies with data size. Furthermore, subsampling provides two different measures for evaluation corresponding to base detector performance and ensemble performance. Note that each individual figure (in Figures 7, 8, and 9) contain 14 boxplots including the base detector. For each of the 14 ensemble methods tested (including the base detector), we computed the number of times the average KNN-detector performed better than LOF, at both \( k = 5 \) and \( k = 10 \). Therefore, there are \( 14 \times 2 = 28 \) comparisons for each data set. For example, the 28 box plots for the LOF performance on glass data set in Figures 7(a) and (c), are compared with the (corresponding) 28 box plots for the average \( k \)-NN detector in Figures 7(b) and (d). We compared both the base detector performance and the ensemble performance. Therefore, we used either the median of the box plot (base detector performance), or the ensemble performance of the averaging combination method. The former is shown in Figure 10(a), whereas the latter is shown in Figure 10(b). A tie was given a credit of 0.5. Note that the sum of the average \( k \)-NN bars and LOF bars must always add up to 28 in each case. What is truly astonishing is that the average \( k \)-NN detector almost always outperforms LOF on the base detector performance, as shown in Figure 10(a).

There were several data sets, where the average \( k \)-NN detector scored a “clean sweep” (28 wins out of 28) irrespective of the subsampling rate or the value of \( k \), which was chosen. Furthermore, the average \( k \)-NN detector also outperforms LOF on the final ensemble performance, although the performance was slightly less dominant in this case. Note that LOF gains a bigger lift from ensembling; however, this lift is often not sufficient to compensate for the poor performance of the base detector.

#### 5.6.1 Is Local Normalization Overrated?

Average \( k \)-NN detectors are generally superior to exact \( k \)-NN detectors, because they are more robust to local noise in the data distribution [4], but this fact has not received sufficient attention in the research community. For this reason, many of the existing comparisons between LOF and unnormalized detectors are often performed using the exact \( k \)-NN detector, which is a suboptimal implementation of unnormalized detectors. Average \( k \)-NN detectors benefit from lower variance. It is noteworthy that LOF also uses (roughly) the average \( k \)-NN distance in its numerator and harmonically averaged \( k \)-NN (in its locality) as its denominator. In this sense, LOF is different from exact \( k \)-NN distance in two ways, one of which is also reflected in the average \( k \)-NN detector. Therefore, it would seem that LOF achieves most of its advantage over the exact \( k \)-NN de-

---

12 A tie on the AUC is an extremely unusual occurrence but it can sometimes occur in smaller data sets, when the number of outliers is small. When the AUC was the same up to at least 4 decimal places, we treated the performance as a tie. This occurred in the case of one of the base detectors on the full data.

13 LOF also uses reachability smoothing.
tector, not from its local normalization in the denominator, but from having a more robust numerator. Indeed, local normalization has many problematic issues of incorporating irrelevant noise from dense regions—a specific example is the \( \infty \)-problem of \( \text{LOF} \). While this specific problem can be fixed by using modifications of \( \text{LOF} \), soft versions of this problem cannot be fixed, when \( k \) data points in very dense regions are very close together by chance. In a dense data set, large values of \( k \) are required for robustness against false positives (because of harmonic normalization), whereas in a sparse data set, smaller values of \( k \) are more appropriate to avoid false negatives. In a data set with widely varying density across different regions, it is impossible to avoid both at a particular value of \( k \). While this problem is also encountered in unnormalized algorithms, local normalization exacerbates this problem more than in unnormalized algorithms. Ironically, the local scheme (\( \text{LOF} \)) has a locality problem in terms of parameter setting. While we recognize that all outliers may not be represented among the rare classes of public data sets, these applications are quite natural. After all, such applications form the primary use-case of unsupervised outlier detection methods when labels are unobserved. Furthermore, because of the typical homogeneity of the relevant causality of outliers in most application-centric settings (e.g., cancer or no cancer), interesting outliers are often global. In such cases, straightforward average \( k \)-NN methods and multivariate extreme value analysis methods (e.g., Mahalanobis method [22]) tend to perform rather well. This strongly suggests that the true benefits of local normalization need to be seriously re-examined by the research community from an application-centric perspective.

### 5.7 Other Implications

The advantages of fixed-rate subsampling are quite limited when using relatively stable detectors such as the average \( k \)-NN detectors. In such cases, the variance reduction lift is quite small, as compared to \( \text{LOF} \). Note that in all the results presented in Figures 7, 8, and 9, the ensemble performance improves over the median more significantly in \( \text{LOF} \), as compared to the average \( k \)-NN detector. This is primarily because of the instability of harmonic normalization in \( \text{LOF} \); as in classification, unstable algorithms are always better for variance reduction. Unfortunately, this instability is also reflected in a poorer overall performance of \( \text{LOF} \), as compared to the average \( k \)-NN detectors. In most cases, this pervasive bias cannot be compensated by better variance reduction. One weakness of fixed-rate subsampling is that it is generally unable to obtain much lift over the median performance with stable detectors. Variable subsampling is still able to obtain a better lift even with stable detectors because it ensembles over more diverse components.

### 6. CONCLUSIONS

In this paper, we present theoretical foundations of outlier ensembles and their applications. The bias-variance theory in outlier detection is almost identical to that in classification. Even though outlier detection is an unsupervised problem like clustering, ensemble analysis in outlier detection is more similar to classification as compared to clustering. In particular, most variance-reduction methods can be adapted easily from classification to outlier detection, although other methods like boosting are more challenging to adapt. We use our theoretical results to design several robust variations of feature bagging and subsampling techniques. We also provide a better understanding of the effectiveness of various combination methods, and propose two new combination methods based on bias-variance theory. The results presented in this paper have the potential to motivate the development of new outlier ensemble algorithms along the lines of well-known classification ensemble algorithms.

### 7. REFERENCES


we experimentally show the invalidity of the arguments for outlier detection ensembles for outlier detection. Therefore, these represent ground-truth scores. Let the corresponding scores output by the outlier detection algorithm be $r_1, \ldots, r_n$. We say that an inversion has occurred if $f_1 < f_2$ and $r_1 < r_2$. In other words, if a data point with a lower probability density (i.e., in a sparse region), has smaller 1-NN distance than a data point in a dense region, then an inversion is assumed to have occurred. Note that this is the key metric that is analyzed in [24]. For each of the $n \cdot (n - 1)/2$ pairs of points in the data set, we computed a non-inversion credit $C(X_i, X_j)$ as follows:

$$C(X_i, X_j) =\begin{cases} 0 & f_i < f_j \text{ and } r_i < r_j \\ 0 & f_i > f_j \text{ and } r_i > r_j \\ 1 & f_i < f_j \text{ and } r_i > r_j \\ 1 & f_i > f_j \text{ and } r_i < r_j \\ 0.5 & f_i = f_j \text{ or } r_i = r_j \end{cases}$$

(8)

The average non-inversion credit $NI(D)$ over all pairs of data points in data set $D$ is defined as follows:

$$NI(D) = \frac{\sum_{i<j} C(X_i, X_j)}{n(n - 1)/2}$$

(9)

In other words, this measure computes the fraction of pairs of points in which the inversion does not occur. Larger values indicate that outliers and inliers will not be inverted. In the ideal case, when no inversions occur, the the value of $NI(D)$ is 1. A value of 0.5 would be expected from a random detector.

Since our primary argument on the effectiveness of subsampling is based on variance, one of the challenges that we faced in our testing was the effect of correlations across multiple ensemble components. Because of the overlaps among the training data sets from various subsamples, the outlier scores (1-NN distances) from various ensemble components are correlated. As a result the variance reduction effects of averaging were curtailed, when the subsamples were large. The problem is that the base data set is finite, and larger subsamples from a base data set always lead to correlated detectors. Correlated detectors generally have a negative effect on any form of bagging or subsampling.

Note that this problem would not be encountered if the base data set were of infinite size. In such a case, the results of any pair of subsamples would be truly independent, and the full effect of variance reduction could be realized. Fortunately, it is indeed possible to simulate such a scenario. In the case of synthetic data sets, the base distribution from which the data set is generated is known, and therefore the subsamples of the desired size can be generated each time from the base distribution. The original base data $D$ is only used to test the outlier scores against each such generated model. Therefore, we generated two different variants of base detectors and ensembles:

1. We constructed the base detectors by drawing subsamples from the original data set $D$. This data set was also used as the test data set, but the 1-NN computation of each point in the test data $D$ was computed only on the subsample of $D$. The average of the 1-NN

### APPENDIX

#### A. INVERSION ANALYSIS

We have already discussed the reasons for the invalidity of the “outlier-inversion argument [24]” in section 3. Here, we experimentally show the invalidity of the arguments for synthetic data sets, which are generated under the same theoretical assumptions of locally uniform distributions.

We used two 1-d locally uniform distributions and a 2-d distribution with clusters of uniformly distributed points. Consider a data set $D$ containing the points $X_1, \ldots, X_n$, with local probability densities $f_1, \ldots, f_n$, which are known from the parameters of the generating distribution. Therefore, these represent ground-truth scores. Let the corresponding scores output by the outlier detection algorithm be $r_1, \ldots, r_n$. We say that an inversion has occurred if $f_1 < f_2$ and $r_1 < r_2$. In other words, if a data point with a lower probability density (i.e., in a sparse region), has smaller 1-NN distance than a data point in a dense region, then an inversion is assumed to have occurred. Note that this is the key metric that is analyzed in [24]. For each of the $n \cdot (n - 1)/2$ pairs of points in the data set, we computed a non-inversion credit $C(X_i, X_j)$ as follows:

$$C(X_i, X_j) =\begin{cases} 0 & f_i < f_j \text{ and } r_i < r_j \\ 0 & f_i > f_j \text{ and } r_i > r_j \\ 1 & f_i < f_j \text{ and } r_i > r_j \\ 1 & f_i > f_j \text{ and } r_i < r_j \\ 0.5 & f_i = f_j \text{ or } r_i = r_j \end{cases}$$

(8)

The average non-inversion credit $NI(D)$ over all pairs of data points in data set $D$ is defined as follows:

$$NI(D) = \frac{\sum_{i<j} C(X_i, X_j)}{n(n - 1)/2}$$

(9)

In other words, this measure computes the fraction of pairs of points in which the inversion does not occur. Larger values indicate that outliers and inliers will not be inverted. In the ideal case, when no inversions occur, the the value of $NI(D)$ is 1. A value of 0.5 would be expected from a random detector.

Since our primary argument on the effectiveness of subsampling is based on variance, one of the challenges that we faced in our testing was the effect of correlations across multiple ensemble components. Because of the overlaps among the training data sets from various subsamples, the outlier scores (1-NN distances) from various ensemble components are correlated. As a result the variance reduction effects of averaging were curtailed, when the subsamples were large. The problem is that the base data set is finite, and larger subsamples from a base data set always lead to correlated detectors. Correlated detectors generally have a negative effect on any form of bagging or subsampling.

Note that this problem would not be encountered if the base data set were of infinite size. In such a case, the results of any pair of subsamples would be truly independent, and the full effect of variance reduction could be realized. Fortunately, it is indeed possible to simulate such a scenario. In the case of synthetic data sets, the base distribution from which the data set is generated is known, and therefore the subsamples of the desired size can be generated each time from the base distribution. The original base data $D$ is only used to test the outlier scores against each such generated model. Therefore, we generated two different variants of base detectors and ensembles:

1. We constructed the base detectors by drawing subsamples from the original data set $D$. This data set was also used as the test data set, but the 1-NN computation of each point in the test data $D$ was computed only on the subsample of $D$. The average of the 1-NN
scores provided the ensemble score. The resulting base detector was referred to as BASE-F and the ensemble detector was referred to as ENSEMBLE-F. The “-F” corresponds to the fact that the base data is finite.

2. In this case, the test data set is fixed to the original data set \( D \), but the subsamples are drawn from an infinite base data set of the same distribution as the test set. This scenario is simulated by generating the subsamples and the test set from the same probability distribution. Note that it is not meaningful to talk of sampling “rates” in this case, because the training data set size is infinite. However, in order to ensure comparability of results with the finite base data, we defined the sampling rate of the subsample with respect to the original (test) data set \( D \). Note that the same test data set \( D \) is used in both finite and infinite sampling. The resulting base detector was referred to as BASE-I and the ensemble detector was referred to as ENSEMBLE-I. The “-I” at the end of the name refers to the fact that subsampling is performed from an infinite data set. Using an infinite base data has the advantage that it allows us to test whether outlier-inlier inversion results for smaller subsamples are indeed true once the effects of correlation between base detectors have been removed.

The results in this section used 300 trials. The accuracy of the base detector is computed by averaging the accuracy over each of these 300 instantiations, whereas the accuracy of the ensemble approach is computed using the averaged 1-NN score of the ensemble.

First, we used a data set \( D \) containing 2000 points drawn from locally uniform distributions in a single dimension. We chose the locally uniform distribution because it approximates the conditions under which the theoretical results of [24] are proposed. The data distribution is shown in Figure 11(a). In this case, the data is distributed in 20 1-d bucket. All 1-d points in the i-th bucket take on uniformly random values in the range \((i,i+1)\). The relative number of points in each bucket is a uniform random variable drawn from \((0,1)\), and it is illustrated on the Y-axis of Figure 11(a). Therefore, the lower bars correspond to regions which are outlier regions in this 1-d data, albeit uniformly distributed. The values on the Y-axis of Figure 11(a), are used as the ground-truth values of \( f \) in Equation 8 for the corresponding data points in that bucket. The 1-NN distance is used as \( r_c \) in Equation 8. The fraction of non-inversions (i.e., \( N(I(D)) \)) of the base system (a 1-NN detector) and ensemble systems both for the case of finite and infinite sampling are illustrated in Figure 11(b). Note that the performance of both base detectors improves with the sampling rate, and no advantage was observed for smaller subsamples. This is because the variance effects dominate, and random draws of smaller subsamples have larger variance. It is noteworthy that this choice of the base detector (absolute k-NN) is the same as the one for which the “outlier-inlier inversion argument” in [24] is constructed. Yet, this inversion was not observed in Figure 11(b). The main improvements were achieved with the use of the variance reduction impact of the ensemble. The ENSEMBLE-F detector did indeed perform quite well for smaller subsamples, but the improvements were achieved because of less correlation among the base components, and therefore better variance reduction. When the subsample size was exactly equal to the size of the full data, no performance improvement was observed because of perfect correlations among the base detectors in ENSEMBLE-F. This is substantiated by the fact that the performance of the ENSEMBLE-I detector improves with increasing subsample size, when the correlations are removed. The gap between the two reflects the gap in variance reduction which arises as a result of increasingly correlated base detectors in ENSEMBLE-F. The performance of ENSEMBLE-I almost always improves with increasing subsample size, which is a result of the statistical effects of using more data. If the outlier-inlier inversion results claimed in [24] had been indeed true, one would expect that to do better at smaller subsamples in ENSEMBLE-I. However, these effects were not observed. We repeated the same experiment with the use of 40 buckets instead of 20 and present the results in Figure 11(c) and (d). The results are very similar to the case of Figures 11(a) and (b). We also tested the effects with 2-d locally uniform distributions of 2000 points. In this case, 30 clusters of uniformly distributed squares were generated, with lower-left corners chosen uniformly at random in \((0,1)\). Each square had a side of length 1/15. The relative number of points in each cluster was a uniform random variable in \((0,1)\), and it represented the ground-truth value of \( f \) in Equation 8. The corresponding scatter plot is shown in Figure 11(e). The corresponding effects on the non-inversion credit with increasing subsample size are illustrated in Figure 11(f). As in the case of the 1-d distributions, the non-inversions reduced with increasing subsample size, and then the performance started reducing because of increasing correlations among detectors. Here, we have also shown the effect of increasing the number of ensemble components in Figure 11(g) and Figure 11(h). The former (Figure 11(g)) is for the case of the 20-bucket 1-d distribution, whereas the latter is for the case of the 2d-distribution. Both the finite and infinite cases are shown in the same plot. It is noteworthy that larger subsamples generally level off sooner and no advantage is observed by increasing the number of ensemble components. Smaller subsamples initially perform poorly, but because of increasing variance reduction, they can often perform better with increasing number of ensemble components. However, there is a limit to this improvement. Subsamples, which are too small, lose too much information in individual detectors to be effective overall, even with a large number of components. For example, at the lowest sampling rate of 0.005, each subsample contained only 10 points, which was not sufficient to meaningfully represent the 20 or 30 clusters. Therefore, the ensemble performance at this sampling rate could not outperform the ensemble performance at higher sampling rates, even after increasing the number of ensemble components. Note that for the case of ENSEMBLE-I, larger subsampling rates almost always provided better performance because the ensemble components were independent, and one could make better use of the greater amount of data. In other words, no outlier-inlier inversion was observed. This is not surprising; the fact that “more-data-is-better” is in tune with the basics of statistics. Clearly, the only significant effect is the variance reduction effect, as in classification.
Figure 11: Effectiveness of base and ensemble on locally uniform data sets (Sampling rates for infinite data set are defined with respect to finite base data set $D$).
Report of the First International Workshop on Learning over Multiple Contexts (LMCE 2014)

César Ferri
Univ. Politéc. de València
Cami de Vera, S/N 46022
València, Spain
cferri@dsic.upv.es

Peter Flach
University of Bristol
Woodland Road
Bristol BS8 1UB, U.K.
peter.flach@bristol.ac.uk

Nicolas Lachiche
ICube, Univ. Strasbourg
300 bd Brant
67400 Illkirch, France
nicolas.lachiche@unistra.fr

ABSTRACT

The first international workshop on Learning over Multiple Contexts, devoted to generalization and reuse of machine learning models over multiple contexts, was held on September 19th, 2014, as part of the 7th European machine learning and data mining conference (ECML-PKDD 2014) in Nancy, France. This short report summarizes the presentations and discussions held during the LMCE 2014 workshop, as well as the workshop conclusions and the future agenda.

1. INTRODUCTION AND MOTIVATION

Adaptive reuse of learnt knowledge is of critical importance in the majority of knowledge-intensive application areas, particularly when the context in which the learnt model operates can be expected to vary from training to deployment. In data mining, the extracted knowledge will generally be used in a new deployment context with different characteristics. Additionally the model might need adaptation, for example if there are manifestations of data shift or concept drift. This has been studied also in machine learning, for example, in relation to variations in class and cost skew in (binary) classification, leading to the development of tools such as ROC analysis for adjusting decision thresholds to operating conditions concerning class and cost skew. More recently, considerable effort has been devoted to research on transfer learning, domain adaptation, and related approaches.

Given that the main business of predictive models is to generalise from the training context to the deployment context, there is clearly scope for developing a general notion of operating context. Without such a notion, a model predicting ‘sales in Prague for this week’ may perform poorly in Nancy for next Wednesday. The operating context has changed in terms of location as well as resolution. While a given predictive model may be sufficient and highly specialised for one particular operating context, it may not perform well in other contexts. If sufficient training data for the new context is available it might be feasible to retrain a new model; however, this is generally not a good use of resources, and one would expect it to be more cost-effective to learn one general, versatile model that effectively generalizes over multiple contexts.

The aim of this workshop was to bring together people working in areas related to versatile models and model reuse over multiple contexts. Given the advances made in recent years on specific approaches such as transfer learning, an attempt to start developing an overarching theory is now feasible and timely, and could be expected to generate considerable interest from the machine learning and data mining communities. Papers were solicited in all areas relating to model reuse and model generalization including the following areas:

- Transfer learning
- Data shift and concept drift
- Domain adaptation
- Transductive learning
- Multi-task learning
- ROC analysis and cost-sensitive learning
- Background knowledge, relational learning
- Context-aware applications
- Incomplete information, abduction
- Meta-learning

The program committee was selected to cover all those areas, with 18 members from 8 countries: Brazil, China, France, Japan, Portugal, Spain, UK and USA.1

The workshop was held at the European machine learning and data mining conference in order to gather specialists from all relevant disciplines. We received submissions from 10 countries: Austria, Bangladesh, Belgium, Brazil, France, Germany, The Netherlands, Romania, Spain and UK. Each paper was reviewed by at least 2 program committee members. 13 regular papers and 3 work-in-progress papers were accepted. Around 40 people participated in the day-long workshop.

1Chowdhury Farhan Ahmed, Univ Strasbourg, France
Charles Elkan, Univ California - San Diego, USA
César Ferri, Univ Politécnica de València, Spain
Peter Flach, Univ Bristol, UK
Amaury Habrard, Univ Jean Monnet, Saint-Etienne, France
Francisco Herrera, Univ Granada, Spain
Meelis Kull, Univ Bristol, UK
Nicolas Lachiche, Univ Strasbourg, France
Dragos Margineantu, Boeing Research, USA
Adolfo Martínez-Usó, Univ Politécnica de València, Spain
Weike Pan, Shenzhen U, China
Ricardo Prudencio, Univ Pernambuco, Brazil
Jaquelin Quiñonero, Facebook, USA
María José Ramírez-Quintana, Univ Politécnica de València, Spain
Carlos Soares, Univ Porto, Portugal
Masashi Sugiyama, Tokyo Institute of Technology, Japan
Bianca Zadrozny, Univ Fluminense, Brazil
Huimin Zhao, Univ Wisconsin – Milwaukee, USA
2. ORGANISATION AND SESSIONS

The technical program of LMCE 2014 consisted of two invited talks and five sessions devoted to a total of 16 contributed papers.

2.1 Invited talks

The workshop started with an invited talk by José Hernández-Orallo (Universitat Politècnica de Valencia, Spain) on Context Change and Versatile Models in Machine Learning. Noting that models are often deployed in a different context than training, the speaker listed many types of contexts and proposed an explicitation of contexts with the corresponding adaptation procedures. His presentation emphasized the importance and usefulness of versatile models that can be adapted to different contexts, in an approach called reframing, with several examples.

Antonio M. Lopez (Universitat Autònoma de Barcelona, Spain) gave the second invited talk to start the afternoon sessions. His presentation entitled Domain Adaptation of Virtual and Real Worlds for Pedestrian Detection concerned autonomous driving, and focused on pedestrian (object) detection. Since labelling examples in images is time-consuming and error-prone, he advocated the use of image synthesis to generate examples. Since going from the training data to deployment data requires domain adaptation, he suggested to use active learning to collect a few examples real examples to add to the many virtual examples. The speaker presented many results on real datasets.

2.2 Cost-sensitive learning

The first regular session dealt with cost-sensitive learning. Ricardo Prudencio presented the first paper, entitled Cost-sensitive measures of instance hardness, by Carlos Melo and himself. He considered the cost of misclassifying an instance, averaged over several learning algorithms, with respect to the cost proportion, under three decision threshold choice methods: score-fixed, score-driven, and rate driven.

The second paper of the session was Classification in Context: Adapting to changes in class and cost distribution by Peter A. Flach. The talk presented known as well as new results on ROC curves and calibration, in particular with respect to F-measure.

Wouter Duivesteijn presented the third paper ROCsearch in a Wider Context – A ROC-Guided Search Strategy for Subgroup Discovery and Beyond by Wouter Duivesteijn, Marvin Meeng and Arno Knobbe. Based on previous work on subgroup discovery published at the 2014 SIAM International Conference on Data Mining, he introduced perspectives for continuous-target subgroup discovery and for exceptional model mining.

2.3 Dataset shift

Meelis Kull presented the first paper, Patterns of Dataset Shift by Meelis Kull and Peter Flach. He described a graph notation for modelling different types of dataset shifts, covering known patterns and introducing new patterns as well.

The second paper of the session was Dataset Shift in a Real-Life Dataset by Chowdhury Farhan Ahmed, Nicolas Lachiche, Clément Charnay and Agnès Braud. Chowdhury Farhan Ahmed demonstrated that the UCI dataset “Washington Bike Sharing” exhibits a range of shifts on input as well as output attributes, depending on the seasons, and suggested to use such dataset to compare existing approaches.

Chowdhury Farhan Ahmed also presented the third paper of this session, Efficient Graph Classification in Shifted Datasets Using Weighted Correlated Feature Selection by Md. Samiullah, Chowdhury Farhan Ahmed, Anna Farinha and Akiz Uddin Ahmed. The authors proposed a correlation-based graph classification approach as well as a new diversity capturing measure.

2.4 Multi-instance, multi-dimensional, multi-task and multi-label learning

The third session gathered four papers that generalise the standard classification setting in some way.

The first speaker was Gitte Vanwinckelen, presenting her work with Hendrik Blockeel. A meta-learning system for multi-instance classification. They evaluated 14 multiple-instance learning techniques on three dataset domains in order to learn a meta-model. Some interesting conclusions were presented, although they need a larger meta-dataset to generalise.

Adolfo Martínez-Usó presented Predictive models for multidimensional data when the resolution context changes by José Hernández-Orallo, Nicolas Lachiche and himself. They considered different aggregation levels in multidimensional data cubes and compared two learning approaches on three datasets.

The third paper of the session was SymReg-MT: Iterative Multi-task Feature Learning Through Weighted Symbolic Regression by Michael Zwick, Holger Schoner and Elsas Rezaie. The authors presented an approach to multi-task feature learning that uses weighting functions based on the frequency of occurrences when looking at the models across different tasks.

The last paper of this session, Multi-label Classification: A Comparative Study on Threshold Selection Methods by Reem Al-Otaibi, Peter Flach and Meelis Kull was presented by the first author. She evaluated the performance on six datasets of three different methods to set the thresholds: global, label-wise and instance-wise.

2.5 Transfer learning

The session started with the paper Multi-System Identification for Efficient Knowledge Transfer with Factored Tensor Recurrent Neural Networks by Sigurd Spieckermann, Siegmund Dull, Steffen Udluft and Thomas Runkler. They proposed a transfer learning approach through parameter transfer to deal with similar dynamical systems and evaluated it on two problems.

Christian Politz presented his paper Subset based Hilbert space projections for transfer learning. He proposed two greedy approaches for selecting data samples from the source in order to improve the transfer learning (using a kernel PCA) when target data and source data follow different distributions.

The third paper was Sequential Decision-Making under Non-stationary Environments via Sequential Change-point Detection by Emmanuel Hadoux, Aurelie Beynier and Paul Weng. The authors considered reinforcement learning using context detection, defined by a drop of quality. They proposed a new algorithm and evaluated it on three domains.

2.6 Applications

The final session of the day concerned applications. Niall Twomey presented the first paper of this session, about a work with Peter Flach on Context Modulation of Sensor Data Applied to Activity Recognition in Smart Homes. They evaluated whether different subsets of sensors lead to different performances and found that light/door sensors are less valuable than motion sensors on the dataset they considered.

The second paper, Identifying dominant models when the noise context is known by César Ferri, José Hernández-Orallo, Adolfo Martínez-Usó and María José Ramírez-Quintana was presented by the last author. This talk was focussed on an experimental study of several regression algorithms according to the noise level. Assuming that the noise level is known would enable to select the best model.
The last paper was *A L2-Norm Regularized Pseudo-Code for Change Analysis in Satellite Image Time* by Anamaria Radoi and Mihai Datcu. The authors proposed and evaluated a new clustering approach to detect changes in a flow of satellite images.

3. CONCLUSION AND DISCUSSION

The workshop concluded with the general impression that it succeeded as a first workshop on learning models over multiple contexts. First, it was a success mainly because of the number and quality of submissions. Secondly, contributions and participants covered a broad range of related areas. Thirdly, there was a broad consensus that those areas are related and that further efforts to clarify their relationships, similarities and differences would be productive.

The open discussion centred around three issues: 1) domain specificities, 2) support for comparing approaches, and 3) continuation. First, participants noticed that some approaches are domain-specific. For example, domain adaptation often deals with images. It would be worth investigating some cross-over, evaluating approaches of one domain on another application domain. It would need pre-processing data, e.g. images, to get a common representation, e.g. attribute-value.

Support for sharing data, algorithms and results would help collaborations. Web platforms such as OpenML.org or CloudFlows.org were suggested. Another way to promote the comparison of approaches would be to set up a challenge for the next workshop. Finally, there was a consensus that it would be worthwhile to continue with the workshop. Most people also thought that it should still be associated to a big data mining or machine learning conference: KDD or ECML-PKDD 2015. The possibility of a journal special issue devoted to the topic of model reuse and context change was also considered, possibly after more authors have had a chance to present their work at the second workshop.

Acknowledgements

The workshop was sponsored by the REFRAME project (http://reframe-d2k.org) and the data mining team of the ICube laboratory (http://icube-bfo.unistra.fr/en/). Further information about the workshop, contributed full papers and presentations can be found at http://users.dsic.upv.es/~flip/LMCE2014/.