Predicting disk failures with HMM- and HSMM-based approaches

Conference Paper · July 2010

DOI: 10.1007/978-3-642-14400-4_30 · Source: DBLP

Siqing Gan
Central South University

Predicting Disk Failures with HMM- and HSMM-Based Approaches

Ying Zhao[1], Xiang Liu[2], Siqing Gan[2], and Weimin Zheng[1]

[1] Department of Computer Science and Technology
Tsinghua University
Beijing, China 100084

[2] School of Mathematical Sciences and Computing Technology
Central South University
Changsha, China 410075

Abstract. Understanding and predicting disk failures are essential for both disk vendors and users to manufacture more reliable disk drives and build more reliable storage systems, in order to avoid service downtime and possible data loss. Predicting disk failure from observable disk attributes, such as those provided by the Self-Monitoring and Reporting Technology (SMART) system, has been shown to be effective. In the paper, we treat SMART data as time series, and explore the prediction power by using HMM- and HSMM-based approaches. Our experimental results show that our prediction models outperform other models that do not capture the temporal relationship among attribute values over time. Using the best single attribute, our approach can achieve a detection rate of 46% at 0% false alarm. Combining the two best attributes, our approach can achieve a detection rate of 52% at 0% false alarm.

Keywords: Disk Failure, SMART data, Hidden Markov Model, Hidden Semi-Markov Model.

1 Introduction

Reliable storage systems serve as one of the fundamental key components for providing reliable performance in large enterprise systems. Although disk failfure is a rare event (lower than 1% annualized failure rate (AFR) reported by vendors [1] and as high as 6% AFR reported by users [2]), it could be very costly in terms of service downtime once it fails. As storage systems become more complex and can easily reach 1000 disks per node (e.g., NetApp FAS6000 series [3]), understanding and predicting failures have been a challenging task for both disk manufacturers (e.g., [1]) and users (e.g., [2]). An accurate failure prediction mechanism is desired to raise warnings to users and to reduce the cost caused by such failures.

The focus of disk failure studies has been on the rate it happens and the relationship between disk failures and observable factors. They can be categorized in two groups. The first group utilizes a broad range of information such as system logs, disk model, room temperature and so on ([2,4,3]), whereas the second

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 390–404, 2010.

group focuses specifically on information collected through the Self-Monitoring and Reporting Technology (SMART) system ([5,6,2]), which provides close monitoring on a number of attributes indicating the health and performance of disks, such as Read Error Rate, Throughput Performance, Seek Error Rate, Reallocated Sectors Count, and so on. The analytical methods on SMART data vary from theshold-based methods (provided by most manufacturers), Bayesian approaches [5], machine learning approaches [6], to correlation study over time [2]. Nevertheless, they all fail to fully consider the characteristics of the observed attributes over time as a time series and tend to make their predictions based on individual or a set of attribute values.

In this paper, we consider observed SMART attributes as time series and employ Hidden Markov Models (HMMs) and Hidden Semi-Markov Models (HSMMs) to build models for "good" and "failed" drives to make predictions. The motivation is that people have observed for some attributes, "a pattern of increasing attribute values (or their rates of change) over time is indicative of impending failure" [6]. It is reasonable to believe that attribute values observed over time are not independent, and a sequence of observed values with certain patterns may be a good indicator on whether or not a drive may fail soon. Hence we would like to fully explore the prediction power of SMART data when treated as time series. The reason we choose HMMs and HSMMs is that they provide flexible, general-purpose models for univariate time series, especially for discrete-valued series, and HSMM is more flexible and offers a large variety of possible temporal dependence structures. In addition to the proposed HMMs- and HSMMs-based methods, we also conduct experiments with the best methods reported in [6], namely the rank-sum test and SVMs. The experimental results show that our approaches improve the prediction accuracy over other approaches for both single- and multiple-attribute settings. Our proposed approaches do not require expensive parameter searching, and are able to reach a detection rate of 52% while keeping the false alarm rate as low as 0%.

The rest of the paper is organized as follows. We first formulate the problem and discuss basic data preparation techniques in Section 2. Section 3 discusses various prediction models for single attributes, including Hidden Markov Models, Hidden Semi-Markov Models, and Rank-sum, whereas Section 4 presents prediction models for multiple attributes, including for combining the results from multiple classifiers and Support Vector Machines. The description of the dataset used in our experiments is in Section 5, as well as experimental methodology and metrics. Section 6 presents the detailed experimental results. Related work is discussed in Section 7, and Section 8 concludes the entire paper.

2 Problem Formulation and Data Preprocessing

2.1 Problem Formulation

Given an attribute matrix $A_{T \times M} = \{a_{ij}\}$ of a disk drive, where a_{ij} is the value of attribute j measured at time i (T is the number of time intervals, M is the number of attributes), the problem of disk failure prediction is to predict whether the disk is going to fail based on $A_{T\times M}$. The attributes could come from a wide range of sources, including room temperature, hours of operating, system logs, and other physical features measured by the Self-Monitoring and Reporting Technology (SMART) system etc.

If the information of a set of failed disk drives and a set of good disk drives is available, the problem of disk failure prediction can be formulated as a classification problem. In particular, let failed disks be positive samples, good disks be negative samples, we can build classifiers based on attribute matrices from both positive and negative samples. When an attribute matrix of a new disk is available, we can use the trained classifiers to classify whether the disk belongs to the positive class (i.e., is going to fail), or the negative class (i.e., is good).

There are three ways to use attribute matrices for obtaining classifiers. First, the entire or part of each row of an attribute matrix can be extracted out as a vector (e.g., at time t, $V_t = \{a_{ti_1}...a_{ti_n}\}$, where $\{i_1, ..., i_n\}$ is a set of attributes of size n), and be used individually as a training sample. In this way, the relationship between attributes is taken into consideration, and this is how SVMs are trained. Second, the entire or part of each column of an attribute matrix can be extracted out as a sequence (e.g., for attribute i, $S_i = \{a_{t_1 i}...a_{t_2 i}\}$ is the attribute sequence from time t_1 to time t_2), and be used individually as a training sample. In this way, the sequence for an attribute is considered as a time series, and the temporal relationship within an attribute is taken into consideration. Our proposed methods in this paper fall into this category. Finally, time series of multiple attributes can be considered at the same time for training, however, it requires complicated learning and may only suitable for a small set of attributes and short period of time.

2.2 Data Preprocessing

Previous research has reported that using binning in data preprocessing is effective in disk failure prediction [5,6]. We employ the equal-width binning procedure is our study, where each attribute's range is divided into a fixed number of bins with equal-width, and attribute values become bin numbers.

3 Predition Models for Single Attributes

In this section, we discuss the two prediction models designed specifically for time series, and the rank sum test, which was reported as the best prediction model for single attributes by [6]. We first briefly describe Hidden Markov Models (HMMs) and Hidden Semi-Markov Models (HSMMs), and how they are used to build prediction models. Then, we briefly review the rank-sum test, and discuss how it is different in terms of using SMART data as time series.

The study of HMMs and HSMMs has a long history. The first HMM work was introduced by the paper of Baum and Petrie [7] in the mid-sixties, and the first application of HSMMs was analyzed in 1980 by Ferguson [8]. They both are built for two stochastic processes: an observed process and an underlying 'hidden' process. Since they provide flexible, general-purpose models for time series, people have been using them on discrete-valued series, categorical series, and many other types of data. The state duration probability density of HMMs is implicitly modeled and follows exponential distribution, which may not be appropriate for some applications. Whereas HSMMs explicitly set a state duration probability density, hence offer a large variety of possible temporal dependence structures.

3.1 Hidden Markov Models

The basic idea of HMM is that the system has an underlying 'hidden' process involving different states and transitions between them at each time t. Each state has probabilities to emit observable symbols, which correspond to the physical output of the system. Given N states $S = \{S_1, S_2, ..., S_N\}$ (the state at time t is denoted as q_t), M distinct observation symbols $V = \{v_1, v_2, ..., V_M\}$, and an observation sequence $O = O_1 O_2 ... O_T$ over time T, the transition probability distribution A, the observation probability distribution B, and the initial state distribution π form the complete parameter set of a model $\lambda = (A, B, \pi)$. In particular, the transition probability distribution $A = \{a_{ij}\}$ is defined as

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], 1 \leq i, j \leq N. \tag{1}$$

The observation probability distribution $B = \{b_{jk}\}$ is defined as

$$b_{jk} = P[O_t = v_k | q_t = S_j], 1 \leq j \leq N, 1 \leq k \leq M. \tag{2}$$

Finally, the initial state distribution $\pi = \{\pi_i\}$ is defined as

$$\pi_i = P[q_1 = S_i], 1 \leq i \leq N. \tag{3}$$

Once λ is determined, we can use the Forward-Backward procedure to calculate the likelihood of observing sequence O from the model. Also, when a set of observation sequences are available, λ can be estimated by the Baum-Welch method or other methods [9].

To apply HMMs to the disk failure prediction problem, N is determined through an automatic model selection procedure, which will be discussed in Section 5. M is the number of bins used in data preparation. We build HMMs from positive training sequences (failed disks) and negative training sequences (good disks), respectively. Then, the testing data (disks without labels) are evaluated using the two models and sequence log likelihood values are calculated accordingly. Finally, we take the difference between the two log likelihood values for each testing sample, and a threshold value is selected to make the prediction (i.e., if the difference is greater than the threshold, the testing disk is predicted as failed, otherwise, it is predicted as good). The threshold value is adjusted to vary the tradeoff between true positive rates and false positive rates.

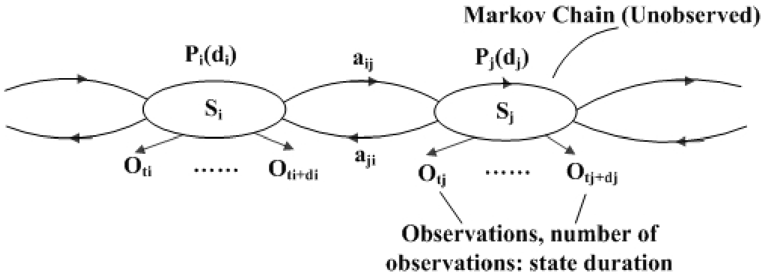

Fig. 1. Illustration of Hidden Semi-Markov Model with specified state duration distributions $p_i(d)$, where multiple observations can associate with one state

3.2 Hidden Semi-markov Models

Letting $a_{ii} = 0, 1 \leq i \leq N$ and setting an state duration probability density, Hidden Semi-Markov Models (HSMMs) allow the underlying 'hidden' process stay in a state for a while, which is a reasonable modification for problems like failure detection, where failures may happen in stages, and the duration of each stage may not follow exponential distributions as implicitly assumed in HMMs.

An illustration of the basic structure of a HSMM is shown in Fig. 1. The complete parameter set for HSMMs now becomes $\lambda = (A, B, D, \pi)$, where the state duration distribution $D = \{p_j(u)\}$ is defined as

$$p_j(u) = P[q_{t+u+1} \neq j, q_{t+u-v} = j, v = 0, ..., u-2 | q_{t+1} = j, q_t \neq j], 1 \leq j \leq N. \tag{4}$$

Sequence likelihood calculation and model parameter estimation can be solved similarly by modifying the Forward-Backward procedure and the Baum-Welch method, respectively [9].

Similarly, we apply HSMMs to the disk failure prediction problem, where N is determined through an automatic model selection procedure, which will be discussed in Section 5. M is the number of bins used in data preparation. We build HSMMs from positive training sequences (failed disks) and negative training sequences (good disks), respectively. Then, the testing data (disks without labels) are evaluated using the two models and sequence log likelihood values are calculated accordingly. Finally, we take the difference between the two log likelihood values for each testing sample, and a threshold value is selected to make the prediction (i.e., if the difference is greater than the threshold, the testing disk is predicted as failed, otherwise, it is predicted as good). The threshold value is adjusted to vary the tradeoff between true positive rates and false positive rates.

3.3 Rank-Sum Test

The Wilcoxon-Mann-Whitney rank-sum is one of the models for comparison against our prediction models, as it was reported as the best model when using single attributes [6]. The rank-sum test [10] is used for testing whether two datasets come from the same distribution. In particular, it is assumed that attribute values measured from good disks follow a "good" distribution, attribute values measured from failed disks follow a "about-to-fail" distribution, and the two distributions are different. Hence, given a reference set R (of size n) consisting of attribute values from good disks and a testing set T (of size m) consisting of attribute values from disks without labels, the rank-sum test statistic W_S can be calculated and used to test against the null hypothesis, i.e., R and T are from the same distribution. We follow the same calculations in performing this test as in [6], where more details about this test can be found.

Note that although a sequence of consecutive attribute values of the test data is used as T for the rank-sum test, it is not used as a sequence in the sense that the testing attribute values are sorted together with reference points with respect to the value, and the temporal dependence among the testing attribute values is ignored.

4 Prediction Models for Multiple Attributes

In this section, we discuss the strategies of combining individual prediction models trained from single attributes. We also briefly discuss Support Vector Machines at the end, which was reported as the best model for multiple attributes in [6].

4.1 Combining Multiple Classifiers

There is a rich literature on the subject of combining multiple classifiers (refer to [11] for a detailed review). Classifiers trained from different training data, feature sets, and classification methods are used together to provide better classification. In this paper, we adopt a simple fixed rule for this purpose, namely, the maximum rule. More complicated combining strategies involving training the combiner can also be considered, however, we do not discuss them here and leave them to future work.

Given M binary classifiers, for one testing sample x, each classifier i returns some sort of confidence value $c_i^+(x)$ for assigning x to the positive class and $c_i^-(x)$ for assigning x to the negative class. In our case, $c_i^+(x)$ and $c_i^-(x)$ are the sequence log likelihoods observed from the positive and negative model, respectively. In general, $c_i^+(x) - c_i^-(x) > 0$ ($c_i^+(x) - c_i^-(x) < 0$) means the testing sample is more likely from the positive (negative) class. Now we have the simple strategy to combine these values.

the maximum rule: The combined value $C(x)$ is defined as

$$C(x) = \max_{i=1}^{M}\{c_i^+(x) - c_i^-(x)\}. \tag{5}$$

The intuitive behind the maximum rule is that we would like to assign x to the positive class (i.e., predicted as failed in our case) if one attribute shows great confidence in doing so. Once we have the combined values, a threshold is used again to trade off between detections and false alarms.

4.2 Support Vector Machines

Support vector machine (SVM) [12] is a state-of-the-art classification technique based on pioneering work done by Vapnik et al. This algorithm is introduced to solve two-class pattern recognition problems using the Structural Risk Minimization principle. Given a training set in a vector space, this method finds the best decision hyperplane that separates two classes. The quality of a decision hyperplance is determined by the distance (referred as margin) between two hyperplances that are parallel to the decision hyperplance and touch the closest data points of each class. The best decision hyperplane is the one with the maximum margin. By defining the hyperplane in this fashion, SVM is able to generalize to unseen instances quite effectively. The SVM problem can be solved using quadratic programming techniques. SVM extends its applicability on the linearly non-separable data sets by either using soft margin hyperplanes, or by mapping the original data vectors into a higher dimensional space in which the data points are linearly separable through appropriate kernel functions. A new example is classified by representing the point the feature space and computing its distance from the hyperplane.

SVM has been applied to a wide range of classification problems because of its many attractive features, including effective avoidance of overfitting, and the ability to handle large feature spaces. The success of SVM has been showed in documents classification and secondary structure predictions. It was also reported as the best classification method on disk failure data when using 25 features from SMART data [6].

5 Datasets and Experimental Methodology

5.1 Dataset

We use the SMART dataset provided by Center for Magnetic Recording Research, University of California, San Diego [6]. The original dataset contains 178 "good" drives and 191 "failed" drives. For each drive, an entry of 60 SMART attributes was recorded every 2 hours for a total of 600 hours. However, failed drives may not survive the entire recording period, thus may have fewer than 300 entries (fewer than 10 entries for some failed drives are observed). In our experiments, since both HMMs and HMMs require non-trivial sequence lengths to be effective, we selected failed drives with more than 50 entries resulting in a total of 99 failed drives. The number 50 is a tradeoff between providing more sequential information to HMMs/HSMMs and keeping enough number of failed drives to fairly compare our results with others. We do not think a minimum length requirement for detecting disk failure is a major limitation, as in practice, the prediction analysis is invoked along with a continuous disk monitoring. It is a safe assumption that before a disk fails, we have already met the length requirement.

In short, our SMART dataset contains 178 good drives and 99 failed drives. We use this dataset for our prediction models and other models for comparison, i.e., the rank-sum test and SVMs. Comparing this dataset with the one used in [6], there are fewer failed disks, making the dataset smaller and more unbalanced (i.e., the ratio between positive and negative class samples is more skewed).

5.2 Metrics

We use receiver operating characteristic (ROC) curves as the metric for evaluating various prediction approaches in our experiments. As mentioned in Section 2, we would like to predict or classify disks with failures accurately through our models. An ROC curve displays the tradeoff between true positive rate and false positive rate. The true positive rate, in our case termed as **detection rate**, is defined as the fraction of positive examples predicted correctly by the model, i.e., the fraction of disks with failures predicted correctly. The false positive rate, in our case termed as **false alarm rate**, is defined as the fraction of negative examples predicted as a positive class, i.e., the fraction of "good" disks wrongly predicted as disks with failures. A good classification model should be as close as possible to the upper left coner of the diagram. Since high false alarm rates are prohibited as the consequent overhead is not acceptable by users (manufacturers' recommendation is 0.1%-0.3%), we only plot ROC curves up to a false alarm rate of 5%, and focus our discussions at the low end (i.e., a false alarm rate less than 1%).

5.3 Experimental Methodology

Now we describe how various models are applied to predict disk failures and how our experiments are set up.

HMMs and HSMMs: The number of distinct observation symbols is the number of bins, which is set to be 10 in our experiments, as suggested by [5,6]. The number of states is determined by an automatic model selection process, in which five models are trained with the number of states varying from 10 to 30, and the one that maximizes the sequence log likelihoods of training sequences is selected. For an attribute, the positive models are trained with sequence segments of the last 50 values of the failed disks in the training set, whereas the negative models are trained with sequence segments of the last 50 values of the good disks in the training set. A disk is predicted as failed if any of its sequence segments of consecutive 50 values over time is predicted as failed. Using a sequence segment of length 100 does not seem to improve performance. In addition, to simply training, we use a parametric state duration distribution instead of non-parametric state duration distribution for HSMMs, i.e., a kernel including a normal distribution is assumed.

We evaluate HMMs and HSMMs in the following way. We randomly selected one fifth of the total failed and good disks as the training sets for positive and negative models, respectively, and use the remaining disks as the testing set. By varying the threshold of the difference of sequence log likelihoods calculated over positive and negative models, we can plot an ROC curve for the trained models.

We repeated this process 5 times, and the average ROC curve was calculated and presented in the following section.

Rank-sum test: We followed the same implementation as in [6]. In particular, for single attribute testing, we randomly choose 50 values from good disks as the reference set (i.e., $n = 50$). Sequence segments consisting of consecutive 15 values of a disk are used as the testing set (i.e., $m = 15$). A disk is predicted as failed if any of its sequence segments over time is predicted as failed by the rank-sum test.

SVMs: We used a 5-fold cross validation for evaluating SVMs. The training samples are attribute vectors from failed disks (positive samples) and good disks (negative samples). A disk is predicted to be failed if any of its attribute vectors over time is predicted as positive. We used mySVM [13] to perform the experiments and followed the same parameter selection as in [6].

6 Experimental Results

6.1 Single Attributes

In the first set of experiments, we focus on the prediction performance of various models when using single SMART attributes. In particular, we started from the 25 attributes identified by [6] with reverse arrangements and z-score test that appear to be promising for distinguishing good and failed disks. We then run our HMM and HSMM predictors and found four attributes provided good failure detection, namely, ReadError18, Servo2, Servo10, and FlyHeight7. We show the ROC curves of HMM, HSMM, and the rank-sum test on these four attributes in Fig. 2 to Fig. 5. Note that ReadError18 and Servo10 were also among the best attributes that appeared to provide good failure detection when using the rank-sum test [6].

The detection rates obtained by the rank-sum test on attribute FlyHeight7 and Servo2 are either very low or too low to be measured, which is consistent with Murray et al.'s observation [6]. On the other two attributes, ReadError18 and Servo10, the HMM and HSMM outperformed the rank sum test significantly. Especially for Servo10, there are no measurable detections at a false alarm less than 1% for the rank-sum test, while HMM and HSMM can achieve a detection rate of 46% and 30% at 0% false alarm, respectively. Except for ReadError18, HMM outperformed HSMM at low false alarm rates. Possible reasons for this observation are two-fold: 1) HSMMs require more parameters to be estimated and our training samples are too limited to produce good training results; 2) to simplify training, we use a parametric state duration distribution, i.e., a kernel including a normal distribution is assumed. Using other distribution families might improve the performance of HSMMs, and we will explore more options in the future.

Note that the performance of the rank-sum test for attribute ReadError18 and Servo10 is slightly different from that reported in [6]. For example, for Servo10,

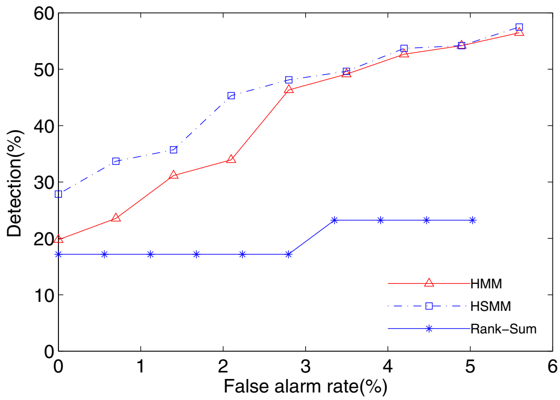

Fig. 2. Performance of HMM, HSMM, and the rank-sum test on disk failure prediction using attribute ReadError18. Results of HMM and HSMM are average over 5 trails.

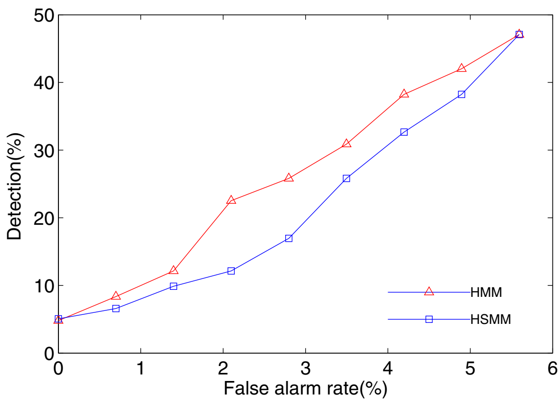

Fig. 3. Performance of HMM, HSMM, and the rank-sum test on disk failure prediction using attribute FlyHeight7. Results of HMM and HSMM are average over 5 trails. No measured detection rates for the rank-sum test with false alarms $\leq 5\%$.

we report a detection rate of 34% and 36% at a false alarm rate of 1% and 2.5%, respectively, while Murray et al. [6] reported a detection rate of 30% and 45%, respectively. There are two reasons behind this observation. Firstly, the reference set of 50 data points is randomly selected from a large population of data points, hence, it is inherently prohibited to reproduce the exact results because of the possible different choice of the reference set. Secondly, our dataset contains fewer failed disks than the one used in [6] as mentioned in Section 5. Nevertheless, we consider the comparison with the rank-sum test has been made carefully, and the conclusion drawn from the comparison is a fair conclusion.

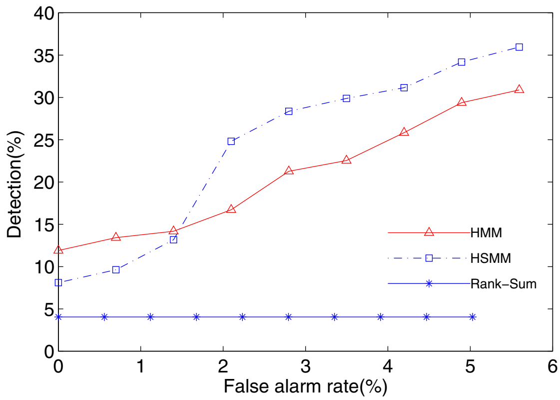

Fig. 4. Performance of HMM, HSMM, and the rank-sum test on disk failure prediction using attribute Servo2. Results of HMM and HSMM are average over 5 trails.

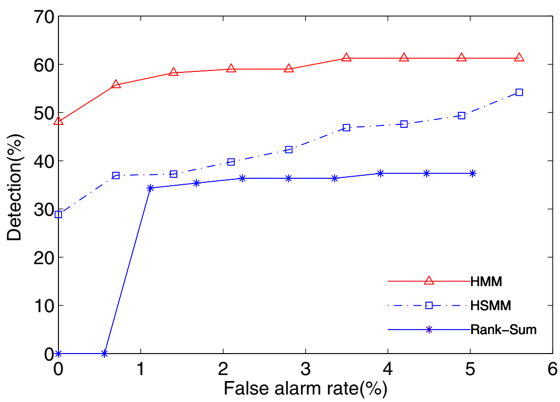

Fig. 5. Performance of HMM, HSMM, and the rank-sum test on disk failure prediction using attribute Servo10. Results of HMM and HSMM are average over 5 trails.

6.2 Multiple Attributes

In the second set of experiments, we focus on the prediction performance of various models when using multiple SMART attributes. From the best single attributes, we choose to combine the best two, namely, ReadError18 and Servo10, which can achieve high detection rates at 0% false alarm. We also compare our model with a SVM trained on 25 attributes. As reported by [6] training SVM on 25 features gave the best multiple-attribute prediction model. We show the ROC curves of the combined HMM using the two attributes and SVM trained on 25 features in Fig. 6.

From Fig. 6, we can see that our combined model with two attributes can achieve a detection rate of 52% at 0% false alarm, and this result is better than the SVM result on 25 attributes. Again, since the dataset used in our experiments is smaller and more unbalanced than the one used in [6], the results of SVM are lower than those reported in [6]. Also there is a huge parameter space to search for finding the best SVM model, and varying parameters like $C, L+, L-$ to trade off between detections and false alarms is not intuitive and hard to interpret. In contrast, our HMM- and HSMM-base approaches employ an automatic model selection process with a much smaller parameter searching space, and varying the threshold of log likelihoods provides an intuitive way to trade off between detections and false alarms.

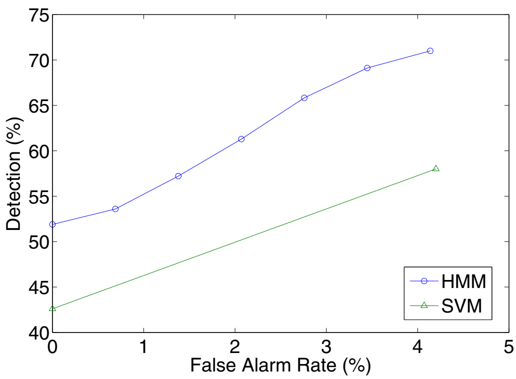

Fig. 6. Performance of the combined HMM and SVM on disk failure prediction. Results of HMM are average over 5 trails. Results of SVM are average over 5-fold cross validation.

6.3 Run Time

All experiments were conducted on a desktop with Pentium 4 dual 2.4GHz CPUs and 1GB memory. We summarize the training and testing time for each prediction model in Table 1. For HMM and HSMM, the single model training time shown in the table is the average for obtaining one model of the positive class or negative class when using one attribute, while the testing time is for calculating log likelihoods of all sequence segments for a testing disk for one model averaged over all testing data. As we mentioned before, HMM and HSMM prediction models have a self model selection procedure, i.e., the number of states are varied for 5 times and the best model is chosen based on the sequence likelihoods of the training data. Hence, obtaining the combined predictor with two attributes involves training $2 \times 2 \times 5$ models (a positive model and a negative model per attribute, 2 attributes, repeated 5 times for model selection) and the combining time is trivial, which results in roughly 1 hour in total. Once we have

Table 1. Summary of run times (in seconds) for training and testing

Prediction Model	Training	Testing (per testing disk)
HMM (singel)	192.4	0.04
HSMM (single)	191.8	0.47
HMM (combined)	3848.4	0.16
Rank-sum	-	2.82
SVM (25 attributes)	1279.1	1.64

the combined predictor, testing one disk needs to be done against 4 individual models, which results in 0.16 seconds for HMM. The testing time of HSMM is much longer than HMM, since we use a normal distribution kernel that demands an integral calculation.

Similarly, the training time for SVM (roughly 20 mins) is for obtaining a SVM model using 25 attributes given one set of parameters. However, to find the optimal set of parameters, an extensive search in the parameter space (i.e., training more than 100 models with different sets of parameters) needs to be done and can be very costly. Note that the rank-sum has the advantage of no training needed; however, testing is a non-trivial process and needs to be repeated for many times with different sequence segments for a testing disk.

7 Related Work

People have been studying disk failures for decades and have been focusing on the relationship between disk failures and observable factors with the hope to predict disk failures accurately and manufacture more reliable disks. These work can be categorized in two groups. The first group utilizes a broad range of information such as system logs, disk model, room temperature and so on ([2,4,3]), whereas the second group focuses specifically on information collected through the Self-Monitoring and Reporting Technology (SMART) system ([5,6,2]), which are more related to our work. Hence we review them in more detail in this section.

Hamerly and Elkan [5] studied SMART data collected from 1934 "good" drives and 9 "failed" drives. They employed supervised naive Bayes learning and mixtures of naive Bayes models to predict "failed" drives, and were able to achieve a detection rate of 30% at a false alarm rate of 0.2%.

Murray et al. [6] collected data from 369 drives (roughly half of which are "good" drives), and studied the performance of multiple-instance naive Bayes, Support Vector Machines (SVMs), unsupervised clustering, the reverse arrangements test, and the rank-sum test on these data. They found that the rank-sum test was the best for single and small sets of SMART attributes (52.8% detection with 0.7% false alarm) and SVMs performed the best for all features (50.6% detection with 0% false alarm).

The work of Pinheiro et al. [2] was conducted on data collected from a system health infrastructure on a large population of hard drives under deployment within Google's computing infrastructure. Their work involved more than one hundred thousand disk drives of different types and periodic SMART data were extracted and cleaned up for their analysis. They found strong correlations between disk failures and some SMART parameters, such as first erros in reallocation, offline reallocations, and probational counts etc. However, they believed that the prediction accuracy based on SMART parameters alone may be quite limited.

HMMs and HSMMs first showed their great success in the context of automatic speech recognition ([9],[8]). Since then for more than two decades, people have been using HMMs and HSMMs on more and more fields involving signal processing applications, such as biological sequences, financial time series and so on. Recently, Salfner and Malek [14] used HSMMs to analysis system logs for accurate error-based online failure prediction on a commercial telecommunication system, where they treat system logs as time series. Along with sequence clustering and noise filtering, their HSMM-based approach was able to achieve a failure prediction F-measure of 0.66.

8 Conclusion

In this paper, we tackle the problem of disk failure prediction from a different angle. We consider various attributes measured at consecutive time intervals for a disk drive as time series, and use HMMs and HSMMs to model such time series to classify "failed" disks and "good" disks. Our proposed prediction models can achieve a detection rate of 46% and 52% at 0% false alarm for single- and multiple-attributes, respectively, which confirms that using time series of attributes is indeed effective in predicting disk failures. Moreover, our combined model does not require expensive parameter searching, and provides an intuitive way of trading off between detections and false alarms.

Acknowledgments. This work was funded by the National Science Fundation China (Project No. 60703058) and the National High-tech R&D Program (863 Program) No. 2008AA01A204 and 2008AA01A201.

References

1. Cole, G.: Estimating Drive reliability in Desktop Computers and consumer electronics systems. Tech. Rep., Seagate Technology Paper TP-338.1 (2000)
2. Pinheiro, E., Weber, W., Barroso, L.A.: Failure Trends in a Large Disk Drive Population. In: 5th USENIX Conference on File and Storage Technologies (FAST 2007), Berkeley, CA (2007)
3. Jiang, W., Hu, C., Zhou, Y., Kanevsky, A.: Are Disks the Dominant Contributor for Storage Failures? A Comprehensive Study of Storage Subsystem Failure Characteristics. ACM Transactions on Storage 4(3), Article 7 (2008)
4. Bairavasundaram, L.N., Goodson, G.R., Pasupathy, S., Schindler, J.: An Analysis of Latent Sector Errors in Disk Drives. SIGMETRICS Perform. Eval. Rev. 35(1), 289–300 (2007)

5. Hamerly, G., Elkan, C.: Bayesian Approaches to Failure Prediction for Disk Drives. In: 18th International Conference on Machine Learning (ICML 2001), Williamstown, MA (2001)
6. Murray, J.F., Hughes, G.F., Kreutz-Delgado, K.: Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application. Journal of Machine Learning Research 6, 783–816 (2005)
7. Baum, L.E., Petrie, T.: Statistical Inference for Probabilistic Functions for Finite State Markov Chains. Annals of Mathematical Statistics 37, 1554–1563 (1966)
8. Ferguson, J.D.: Variable Duration Models for Speech. In: Symposium on the Application of Hidden Markov Models to Text and Speech, Princeton, New Jersey, pp. 143–179 (1980)
9. Rabiner, L.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. IEEE Transactions on Information Theory 77(2), 257–284 (1989)
10. Lehmann, E.L., D'Abrera, H.J.M.: Nonparametrics: Statistical Methods Based on Ranks. Prentice Hall, Upper Saddle River (1998)
11. Duin, R.P.W.: The Combining Classifier: to Train or not to Train? In: 16th international conference on Pattern Recognition, Quebec City, Canada, pp. 765–770 (2002)
12. Vapnik, V.: Statistical Learning Theory. John Wiley, New York (1998)
13. `http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM`
14. Salfner, F., Malek, M.: Using Hidden Semi-Markov Models for Effective Online Failure Prediction. In: 26th IEEE Symposium on Reliable Distributed Systems (SRDS 2007), Beijing, China, pp. 161–174 (2007)