6
Conclusion

In this work, we considered the problem of anomaly detection in multiple co-evolving streams of numerical data. Our target domain is a data center, where huge volume of real-time data needs to be monitored by the operations team with the objective to maintain high-availability and quality of the services. Traditional monitoring solutions rely on experts to configure naïve thresholding on single streams, which is not efficient and does not explore the inherent correlated nature of the data.

We were inspired by promising published results using the SPIRIT algorithm [Hoke et al., 2006], which associates anomalies to changes in the latent dimension by tracking the principal subspace incrementally. However, we observed it is an extension to the PASTd algorithm, ergo the orthonormality of the subspace basis is falsely assumed. A re-orthonormalization step implies a computational complexity of $\mathcal{O}(Nr^2)$ per update and not $\mathcal{O}(Nr)$ as previously advertised. In the worst case analysis, even though unlikely, where $r = N$ this algorithm offers no advantage over a batch SVD on the exponentially updated covariance matrix with exact subspace estimates. We also learnt that KOAD, the other alternative from the literature based on kernel-RLS, is unstable numerically and fails in our experiments when different scales of values are used and normalization is not performed.

6.1
Contributions

In the light of these issues, we contribute a new rank-adaptive algorithm for fast principal subspace tracking with a true dominant complexity of $\mathcal{O}(Nr)$. FRAHST is our extension to the recent recursive row-Householder method [Strobach, 2009a], which is the state-of-the-art for this class of subspace trackers: it is numerically robust and provides excellent principal subspace estimates. Our extension adapts the dimension of the tracked subspace in real-time by discovering the number of latent variables necessary to guarantee a given reconstruction accuracy. We demonstrated how the $\mathcal{O}(r^3)$ number of flops from the original algorithm can be reduced to $\mathcal{O}(r^2)$ by exploiting

recurrent rank-one updates in a pair of QR factors, thus enabling FRAHST to achieve a total computational complexity of $5Nr + \mathcal{O}(r^2)$ flops per update. The space requirements are similarly small and do not depend on the time t.

We compared our technique with other two online and two batch algorithms for anomaly detection in four different datasets, and it achieved overall excellent performance being the only algorithm with F_1 score equal or greater than 80% in four experimented datasets. We showed how embedding lagged values in the input vector allows temporal correlations to be captured by a pre-processing step that can be easily performed online. It effectively allowed a subtle broken correlation to be detected in two network links, corresponding to a serious failure caused by a telecom operator but which was not discovered by the traditional mechanisms when it occurred.

A real-time system was implemented in a data center and visualization of FRAHST's latent variables along with the changes in rank provides a good synopsis of the monitored streams with a natural interpretation. The adopted system was presented as a use case at an international forum [Clemente and Vieira, 2009] and promoted as being a powerful tool to have when operating a data center. Our work will also be presented at the ACM SAC next year Teixeira and Milidiú [To Appear], where we expect to reach a wider audience.

The results indicate that a robust subspace tracker is well suited for spotting anomalies in streaming data of low intrinsic dimension, even when compared to algorithms that can look at the entire dataset more than once. FRAHST was consistently better than SPIRIT in all criteria, hence we can safely recommend it for the same tasks SPIRIT has been used in the literature, such as forecasting and pre-processing for other machine learning algorithms.

6.2 Future work

More recently, the role of the Householder transformation was formalized in the Householder Compressor Theorem [Strobach, 2009b] and the author offered an algorithm based on a very fast recurrent rank-one updates of a square-root triangular factorization of the S-matrix. These optimizations can be easily incorporated into FRAHST for anomaly detection in order to reduce the total number of flops per update.

Our algorithm depends on intuitive parameters: the forgetting factor α and desired reconstruction error $1 - f_E$. For all our datasets settings values similar to the ones used in the literature were sufficient to achieve excellent results. But we acknowledge that they might vary accordingly to the data, and it would be much better to have a totally parameter-free algorithm and we

reckon it is possible. Ideas from adaptive memory from adaptive filtering might be applied; and more importantly, new results stemming from random matrix theory concerning the Tracy-Widom distribution seem to offer automatic ways to calculate optimal dynamic thresholds under clear optimization formulations [Perry and Wolfe, 2009].

Other techniques to detect anomalies may be combined with our subspace tracker, in particular we highlight [Chen et al., 2007] where a sequential EM algorithm learns the distribution of two chosen statistics and significant changes in the probability density indicate potential anomalies. However, there are still challenges to implement it efficiently under the streaming constraints.

Even the latent variables often have patterns, like weekly scheduled backup of databases. This patterns should be compressed and incorporated into the current model to reduce false positives. One alternative is to investigate how to learn auto-regressive models for each latent variable in an online manner, and this might include using established multi-scale data structures like Wavelets [Milidiú et al., 1999].

Another interesting direction is to embrace supervised or semi-supervised learning, which would eventually allow anomaly prediction. There is currently undergoing interesting work being done at IBM [Gu et al., 2008a, Gu and Wang, 2009, Gu et al., 2008b], where both Markov models and decision trees were adapted to the streaming scenario and show promising ideas for predicting faults in complex systems.

We are also currently interested in implementing our algorithm to run in GPUs[1] or in FGPAs[2]. This would eventually allow very efficient processing of thousands of streams at very high rates.

6.3 Final words

Stream computation and learning over data streams seem a natural evolution of information technology given the dynamic nature of most data. Our work touches many important subjects such dimensionality reduction, rank estimation and anomaly detection under the streaming constraint and offers a useful *any-time* fast method for learning patterns in multiple streams of data.

[1]Graphics processing unit
[2]Field Programmable Gate Array