ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Random Projection in supervised non-stationary environments

Moritz Heusinger[1] and Frank-Michael Schleif[1] *

1- UAS Würzburg-Schweinfurt - Department of Computer Science
Sanderheinrichsleitenweg 20, Würzburg - Germany

Abstract. Random Projection (RP) is a popular and efficient technique to preprocess high-dimensional data and to reduce its dimensionality. While RP has been widely used and evaluated in stationary data analysis scenarios, non-stationary environments are not well analyzed. In this paper we provide a profound evaluation of RP on streaming data. We discuss how RP can be bounded for streaming data using the Johnson-Lindenstrauss (JL) lemma. In particular we analyze the effect of *concept drift*, as a key challenge for streaming data. We also provide experiments with RP on streaming data, using state-of-the-art streaming classifiers like *Adaptive Hoeffding Tree*, to evaluate its efficiency.

1 Introduction

A common problem in data analysis tasks, like classification is the high dimensionality of the data. Given the data is intrinsically low dimensional, the dimensionality can be reduced. To address this problem, there exist different projection and embedding methods. The most common projection technique is the Principal Component Analysis (PCA). But also embedding techniques like Locally Linear Embedding, ISOMAP, Multidimensional Scaling and t-distributed Stochastic Neighbour Embedding (t-SNE) are widely used [14].

The problem of these methods is, that they are often working on high-dimensional but not very high-dimensional data, e.g. thousands of input dimensions. Also the PCA is costly taking $\mathcal{O}(d^2n + d^3)$, where n is the number of samples and d the number of features. Hence, less expensive dimensionality reduction techniques are desirable. This points to Random Projection (RP), which is a dimensionality reduction method to reduce the dimensionality of data from very high to high [1]. The guarantees given by Random Projection are based on the JL lemma [9], which is also used to determine a suitable number of low dimensions where the projection does not make an error greater than ϵ.

In this paper we provide a comprehensive study on RP in non-stationary environments by comparing classifier performance on projected and original data streams, additionally concept drift (CD) is taken into account. Finally, we review the Johnson Lindenstrauß lemma in the presence of not knowing the length n of a data stream.

*Supported by StMWi, project *OBerA*, grant number IUK-1709-0011// IUK530/010.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

2 Related Work

In stationary environments as well as on streaming data, dimensionality reduction is a topic of interest, which allows data visualization and reduces the complexity of algorithms. Especially in streaming environments this is important because of the limited amount of time to predict label y for an incoming data point $\mathbf{x}$ [2].

Some dimensionality reduction algorithms have been adapted to work with streaming data. In [7] an online PCA for evolving data streams has been published, which only keeps track of a subspace of small dimensions that capture most of the variance in the data.

Also, Random Projection has already been applied in the fields of non-stationary data, but without theoretical foundation or analysis. It was also not analyzed or proven for non-stationary environments w.r.t. JL lemma. A stream clustering algorithm called streamingRPHas, which uses RP and locality-sensitivity hashing was published in [5]. Another work uses a Hoeffding Tree ensemble to classify streaming data, on a lower dimensional space, obtained by RP [12]. Both approaches use RP in streaming context without taking the effect of CD on RP into account, which is a major challenge in supervised streaming analysis.

So far a comprehensive study on RP and the envolved particularities of streaming data is missing, addressed in this experimental study.

3 Preliminaries

3.1 Random Projection

Consider a data set $\mathbf{X} \in \mathbb{R}^{n \times d}$ with datapoints $\mathbf{x}$ in a d-dimensional space. The data is projected to a lower dimensional space $k \ll d$ by a function

$$\pi(\mathbf{X}) = \mathbf{R} \times \mathbf{X} \tag{1}$$

via a random matrix $\mathbf{R} \in \mathbb{R}^{k \times d}$. The distortion introduced by a random projection π is asserted by the fact that π defines an ϵ-embedding with high probability defined by JL:

$$(1-\epsilon)\|\mathbf{u}-\mathbf{v}\|^2 < \|\pi(\mathbf{u})-\pi(\mathbf{v})\|^2 < (1+\epsilon)\|\mathbf{u}-\mathbf{v}\|^2 \tag{2}$$

where $\mathbf{u}$ and $\mathbf{v}$ are rows taken from data with n samples and d dimensions [1]. The parameter k is determined by JL lemma

$$k >= \frac{4 \log n}{(\epsilon^2/2 - \epsilon^3/3)} \tag{3}$$

with an error less than ϵ [1]. There exist different approaches for generating a sparse random projection matrix $\mathbf{R}$. One approach creates $\mathbf{R}$ by drawing samples from $\mathcal{N}(0,1)$ and is called Gaussian RP (RP-G). The other approaches

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

create $\mathbf{R}$ with entries $r_{i,j}$ where i denotes the i-th row and j the j-th column of $\mathbf{R}$, mainly differing in underlying density distribution:

$$r_{i,j} = \sqrt{s} \begin{cases} -1 & \text{with probability } \frac{1}{2s} \\ 0 & \text{with probability } 1 - \frac{1}{s} \\ +1 & \text{with probability } \frac{1}{2s} \end{cases} \tag{4}$$

In [1] it is recommended to use a density of $s = 3$. In [10] it is recommended to use $s = \sqrt{d}$ instead, because this speeds up the computation by having a more sparse matrix $\mathbf{R}$. The first method is often referred to as sparse RP (RP-S) while the latter is called very sparse RP (RP-VS). Calculation and usage of $\mathbf{R}$ costs $\mathcal{O}(dkn)$ and $\mathcal{O}(ckn)$ when $\mathbf{X}$ is sparse, having c non-zero values per column [4].

3.2 Stream Analysis

Many preprocessing algorithms can not be applied to streaming data, because there are some particularities [2] when analyzing data in non-stationary environments. Main challenges are (1) inspection at a time, (2) time and memory constraints, (3) dynamics of the data stream in particular CD, (4) CD detectors are very slow in high dimensions. Some algorithms have been adapted to work with streaming data with a comprehensive review in [2]. In particular, statistics cannot be calculated over the whole data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t\}$ because the data become available once at a time, asking for a flexible model. Thus, when applying RP we will project one data point $\mathbf{x}_t$ at time t by our initially created projection matrix $\mathbf{R}$ to provide the projected data set to an online learning algorithm. Another problem in non-stationary environments is CD which means that joint distributions of a set of samples $\mathbf{X}$ and corresponding labels $\mathbf{y}$ between two points change in time:

$$\exists \mathbf{X} : p(\mathbf{X}, \mathbf{y})_t \neq p(\mathbf{X}, \mathbf{y})_{t-1} \tag{5}$$

The term virtual drift refers to a change in distribution $p(\mathbf{X})$ for two points in time. Furthermore, we can rewrite Eq. (5) to

$$\exists \mathbf{X} : p(\mathbf{X})_t p(\mathbf{y} \mid \mathbf{X})_t = p(\mathbf{X})_{t-1} p(\mathbf{y} \mid \mathbf{X})_{t-1} \tag{6}$$

For a comprehensive study of various CD types see [6].

For this case online classification algorithms use statistical tests to detect CD. These tests are adapted and used as so-called CD detectors (e.g. Adaptive Windowing [3] and Kolmogorov-Smirnov Windowing [13]).

4 Random Projection in non-stationary environments

Assume we have a stream with a fixed number of dimensions, which will not change over time, we can still use Eq. (1).

Due to the fact, that we depend on a fast projection in non-stationary settings, the very sparse approach of [10] seems to be most suitable. The matrix

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

creation using RP-VS costs $\mathcal{O}(ck)$ with c nonzero entries per column and can be done in advance. The JL lemma shows, that k does not depend on d, instead k depends on number of instances n [9], which can be addressed by a window technique.

Determining a suitable number of dimensions for RP is a problem in the field of streaming data. When the size of the whole data set n is known, we can calculate a suitable number of dimensions by Eq. (3). However, a stream has potential infinite length and unknown fixed n, hence a window of $|\mathbf{W}|$ samples can be used, with k as defined by JL lemma. The window size of HT [3] and SAM-KNN [11] (both without RP) is often chosen around 1,000. Hence, we use a sliding window $\mathbf{W}$ of size 1,000 over a data stream and assume that we are learning and predicting a single datapoint at every timestep t. Now we only need to preserve the distances of 1,000 datapoints, because we are only calculating distances on $\mathbf{W}$ instead of $\mathbf{X}$. To find a suitable number of dimensions k by Eq. (3) we have to choose a value for the introduced distortion ϵ. A good choice here is $\epsilon = 0.2$ [4]. This leads us to a bound of $k_W = 1594$ for a fixed length window $|\mathbf{W}| = 1,000$. E.g. at the beginning of a stream, the window is not completely filled, however this is not a problem, because having $n < |\mathbf{W}|$ samples leads to $k \leq k_W$.

5 Experiments

As benchmark classifiers we use Adaptive Robust Soft Learning Vector Quantization (ARSLVQ), which is a version of RSLVQ optimized for handling evolving data streams proposed in [8], SAM-KNN [11] and Hoeffding Adaptive Tree (HAT) [3] as state-of-the-art streaming classifiers[1].

In our first experiment we want to determine, which RP technique is more suitable for streaming data. Based on the theory, RP-VS should be the fastest method [10], but we cross check by the accuracy of our classifier, to analyze classification performance on the separate projections. In this setting we are projecting 10,000 dimensions to k_W dimensions. Timed in seconds, RP-S took 82.67 ± 0.69, RP-VS 5.06 ± 0.01 and RP-G 36.27 ± 1.06, thus RP-VS performs best, as we have already expected by one magnitude. Also, an NN classifier achieved an accuracy ~ 100 % for each projection on 10,000 Gaussian samples.

In the next step we want to show whether it is preferable for stream classifiers to use the projected instead of original data, by comparing the runtime of classifying the high dimensional Gaussian data vs. the lower dimensional data provided by an RP algorithm. We are including the projection time into the runtime.

As shown in Table 1 the runtime of all classifiers has improved by operating on the projected space over predicting the original 10,000 dimensional data without suffering accuracy.

In our next experiment we are using common data stream generators and real data to explore how RP can be used on data containing drift and being not i.i.d.

[1] Experiments implemented in scikit-multiflow https://scikit-multiflow.github.io/

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Classifier	Original	RP-S	RP-VS	RP-G
RSLVQ_{ADA}	251.49 ± 0.14	210.25 ± 0.50	$\mathbf{129.88 \pm 0.69}$	244.54 ± 0.29
SAM-KNN	7313.82 ± 22.99	$\mathbf{1106. \pm 45.46}$	1219.61 ± 14.86	1205.40 ± 218.75
HAT	8872.95 ± 20.82	3004.79 ± 12.01	$\mathbf{2890.46 \pm 4.89}$	3445.05 ± 2.05

Table 1: Runtime comparison for the classifiers and RP methods between the original data and the projected data. Best performance marked bold.

In this scenario we will only use GP-VS [10] on 50,000 samples. Commonly used stream generators do not contain high dimensional data. Thus, we are enriching every stream generator with random features until it has 10,000 dimensions, and reducing the dimensions to k_W via RP. We used the STAGGER and the SEA generator [6]. When using gradual drifts, the gradual drift will take place at iteration 25,000 by a width of 10,000 indicated by $(.)_G$. Abrupt drift will also take place at sample 25,000 indicated by $(.)_A$. Reuters dataset[2] is projected from original 4,773 features to k_W. Abrupt drift is introduced at sample 1,237, last sample is 2,445.

Algorithm Dataspace	RSLVQ_{ADA} Original	RSLVQ_{ADA} Projected	SAM-KNN Original	SAM-KNN Projected	HAT Original	HAT Projected
STAGGER_G	62.22 ± 2.29	64.26 ± 3.42	38.94 ± 11.24	38.97 ± 11.05	97.61 ± 4.77	67.90 ± 3.17
STAGGER_A	62.98 ± 3.14	61.51 ± 3.99	38.74 ± 11.11	38.84 ± 11.20	41.32 ± 16.21	68.39 ± 4.05
SEA_G	78.97 ± 4.16	78.12 ± 5.79	25.16 ± 1.45	25.29 ± 1.45	62.14 ± 1.31	74.41 ± 1.64
SEA_A	76.35 ± 4.20	70.20 ± 5.76	39.76 ± 1.70	39.63 ± 1.80	52.36 ± 0.80	68.53 ± 4.40
Reuters	99.64 ± 0.00	99.47 ± 0.00	99.51 ± 0.00	99.47 ± 0.00	oom	85.03 ± 0.0
STAGGER_G	114 ± 0	105 ± 1	5273 ± 8219	1320 ± 1565	7798 ± 2716	1754 ± 400
STAGGER_A	115 ± 0	108 ± 1	6198 ± 10125	1489 ± 1885	9859 ± 699	1102 ± 240
SEA_G	114 ± 0	104 ± 1	12809 ± 2439	3215 ± 1373	6747 ± 8	1562 ± 30
SEA_A	114 ± 0	106 ± 1	7639 ± 2410	604 ± 31	6519 ± 18	1725 ± 353
Reuters	3.69 ± 0.01	1.22 ± 0.03	156.28 ± 7.05	33.34 ± 0.13	oom	71.83 ± 0.17

Table 2: Comparison of concept drift streams w.r.t. accuracy and runtime

The upper four rows of Table 2 show the results w.r.t. accuracy. We can see, that for RSLVQ and SAM-KNN there is no relevant difference in the accuracy comparing high vs low dimensional streams. The reason that some algorithms perform poor even in the high dimensional space is, that every stream has been enriched with random noise to get 10,000 features. HAT often performs better in the projected space, but on STAGGER_G some information is lost. However, it does not seem to make a difference in general if we are handling CD streams in projected or in original space as the other results of Table 2 show. The runtime complexity is however strongly effected. HAT could not be evaluated on original Reuters due to memory issues, using 32 GB (out of memory - oom). The lower four rows of Table 2 show runtime results. We see, that the runtime on the projected space is better on every stream, independent of algorithm usage. However, the prototype classifier only saves a small amount of runtime, thus the trade-off may not be desirable depending on the stream, this is the case because RSLVQ lost some of its accuracy performance on SEA_A. The runtime savings for the NN approach and the tree-based approach is ~ 75 % which seems to be very good, expect for HAT on STAGGER_G generator because of the enormous trade-off in accuracy.

[2]https://github.com/ChristophRaab/stvm, we are using 'org vs people'.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

6 Outlook & Conclusion

In future work we want to consider the case, that features of a data stream will be removed or new features are added, e.g. because a sensor gets replaced. Hence, the random matrix $\mathbf{R}$ needs to be adapted. More recent stream classifiers use advanced statistical CDD which are expected to perform better in low dimensional space. In summary our experiments have shown, that using RP in non-stationary environments can save a lot of time on some classifiers. While the RSLVQ$_{ADA}$ shows a moderate improvement, the effect is strong in case of other state-of-the art methods like SAM-KNN and HAT. Our results show, that applying RP has no negative impact on the accuracy. Thus, the usage of RP on streaming data leads to enormous runtime savings and the error is bounded on window-based classifiers.

References

[1] Achlioptas, D.: Database-friendly random projections: Johnson-lindenstrauss with binary coins. Journal of computer and System Sciences 66(4), 671–687 (2003)

[2] Aggarwal, C.C.: A survey of stream classification algorithms. In: Data Classification: Algorithms and Applications (2014)

[3] Bifet, A., Gavaldà, R.: Adaptive learning from evolving data streams. In: Advances in Intelligent Data Analysis VIII, IDA 2009. Proceedings. Lecture Notes in Computer Science, vol. 5772, pp. 249–260. Springer (2009)

[4] Bingham, E., Mannila, H.: Random projection in dimensionality reduction: Applications to image and text data. In: Proceedings of the Seventh ACM SIGKDD. pp. 245–250. KDD '01, ACM, New York, USA (2001)

[5] Carraher, L.A., Wilsey, P.A., Moitra, A., Dey, S.: Random projection clustering on streaming data. In: 2016 IEEE 16th ICDMW. pp. 708–715 (2016)

[6] Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Computing Surveys 46(4), 1–37 (2014)

[7] Grabowska, M., Kotłowski, W.: Online principal component analysis for evolving data streams. In: Computer and Information Sciences. pp. 130–137. Springer (2018)

[8] Heusinger, M., Raab, C., Schleif, F.M.: Passive concept drift handling via momentum based robust soft learning vector quantization. In: Advances in SOM, LVQ, Clustering and Data Visualization. pp. 200–209. Springer (2020)

[9] Johnson, W.B., Lindenstrauss, J.: Extensions of lipschitz mappings into a hilbert space. Contemporary mathematics 26(1), 189–206 (1984)

[10] Li, P., Hastie, T.J., Church, K.W.: Very sparse random projections. In: Proceedings of the 12th ACM SIGKDD. pp. 287–296. ACM (2006)

[11] Losing, V., Hammer, B., Wersing, H.: KNN classifier with self adjusting memory for heterogeneous concept drift. Proc. - IEEE, ICDM pp. 291–300 (2017)

[12] Pham, X.C., Dang, M.T., Dinh, S.V., Hoang, S., Nguyen, T.T., Liew, A.W.: Learning from data stream based on random projection and hoeffding tree classifier. In: 2017 International Conference on Digital Image Computing (DICTA). pp. 1–8 (2017)

[13] Raab, C., Heusinger, M., Schleif, F.M.: Reactive soft prototype computing for frequent reoccurring concept drift. In: Proc. of the 27. ESANN. pp. 437–442 (2019)

[14] Sacha, D., Zhang, L., Sedlmair, M., Lee, J.A., Peltonen, J., Weiskopf, D., North, S.C., Keim, D.A.: Visual interaction with dimensionality reduction: A structured literature analysis. IEEE Trans. Vis. Comput. Graph. 23, 241–250 (2017)