Dimensionality reduction in (large) measurement datasets

***Abstract*—Modern network measurements may contain hard to discern facets. Dimensionality reduction is one method to emphasize and reduce features that are hard to find in raw data, 2D plots or summary statistics.**

This is because network features are often derived from basic, measurable entities, for example, quality might be a function of the delay, loss, access technology or the environment. Some features may be coupled, for example network load and server response times may be related in complex ways. Consequently, we evaluated four dimensionality reduction techniques based on clustering: Principal Component Analysis, Kernel PCA, Linear Discriminant Analysis, and t-Distributed Statistical Neighbour Embeddings.

We measured network attributes during the most-viewed boxing match streamed from three globally accessible servers. We measured the network and server delays, the packet loss. The access technologies were WiFi and Ethernet from two environments, home and work. The four algorithms were evaluated in terms of clustering or separation visualisation and CPU and memory performance.

***Index Terms*—Machine learning, measurements, dimensionality reduction.**

I. Introduction

Dimensionality reduction has been widely applied in many fields. Reduction techniques have been used to lower the amount of data from the original dataset and leave only the statistically relevant components in the (output) processed data. Dimension reduction also improves the performance of classification algorithms, as it removes noisy irrelevant data points. It also allows for improved visualisation of complex measurement data sets. In the ML community, dimensionality reduction is also known as feature extraction/selection.

In large sets of network data, we also want to quantify the major statistical contributors and their interaction. This interaction can be broadly considered as the covariance between the measured entities. In a networked setting, covariance can arise as coupled delays, which we will explore in Section III-A.

Measurement data typically has many thousands of data points, but not many explicit features. Implicit features, often hidden, such as the network load are embedded into the measurements. The captured environment constitutes an important feature of networked measurements. In contrast, the number of features in image processing can be in the tens of thousands, but fewer implicit features and data points.

Our contribution is a performance and visualisation evaluation of four dimensionality reduction techniques using measurement data from a popular sporting event.

Method	Type	Parametric	Parameters	Compl.	Mem.
PCA	Linear	No	-	$O(D^3)$	$O(D^2)$
kPCA	Non-lin.	Yes	$k(\cdot,\cdot)$	$O(N^3)$	$O(N^3)$
LDA	Linear	No	-	-	-
t-SNE	Non-lin.	Yes	$Perp(\cdot)$	$O(N^2)$	$O(N^2)$

TABLE I: D is the dim. of the measurement space, N is the number of data points. $P = O(ND + MT + DT)$, where $T = min(N, D)$, and $k()$ is a kernel function.

II. Four techniques

A. Rationale and evaluation

We chose PCA as it is ubiquitous in dimensionality reduction, it is computationally efficient, linear and parameterless. It has many variants, kernel, probabilistic, discriminant, see the PCA-specific book [1]. PCA essentially works by separating points as far as possible based on the highest varying field. Kernel PCA performs analysis in the high dimensional space using a kernel function to find the principal components, see Table I. LDA is also closely related to principal component analysis and factor analysis in that they both look for linear combinations of variables that best explain the data. t-SNE uses a non-linear clustering algorithm, which has proved popular in image analysis. We devote a little more space to t-SNE in how it works [2].

B. Principal Component Analysis (PCA)

PCA is a linear variance-based method for reducing the dimensionality of data. It uses an orthogonal transformation to convert a dataset of correlated variables, into a set of sorted values of linearly uncorrelated variables, called the principal components [3]. The transformation ensures the 1st component has the largest variance, the 2nd the next highest variance, and so on. The number of principal components will always be less than (or equal) to the number of original features or measurement attributes. Dimensionality reduction allows features that contribute little to the dataset to be removed. A visualisation is shown in Figure 1 (left).

C. Kernel PCA (kPCA)

Is an extension of PCA using linear algebra's kernel methods. A kernel, known as the nullspace, is the set of vectors in the domain of the mapping which maps to the zero vector. They are key in inverting matrices, a fundamental linear

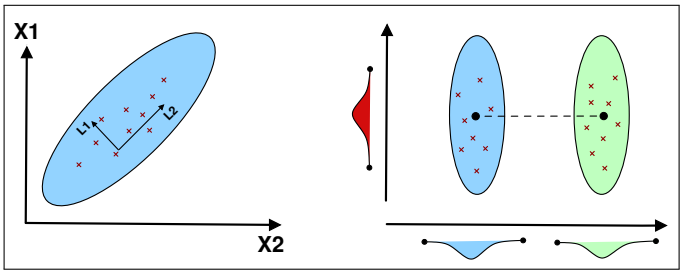

Fig. 1: Left: In PCA the component lengths represent their relative contributions, and the angle between them their correlation see also Figure 4. PCA tries to maximise the component axes. Right: LDA tries to maximise the axes separation for class separation.

algebra operation. Formally, let $T : V \rightarrow W$ be a linear transformation between vector spaces:

$$ker(T) = T^{-1}(0) = \{v \in V | Tv = 0\}.$$

Kernels are used in SVMs, popular in machine learning and in kernel density estimates, and a continuous function representing a discrete histogram. Notably, kPCA nonlinearises the linear dimensionality reduction method by calculating the images of the pairs of data rather than the real values, saving computation, known as the kernel 'trick'. That said, the kernel has to be chosen carefully, to prevent variations in the data appearing as the same, which cannot happen in PCA as the eigenvalues are used to rank the eigenvectors based on their variation in vanilla PCA.

D. Linear Discriminant analysis (LDA)

LDA is also closely related to principal component analysis (PCA) and in that it looks for linear combinations of variables that best explain the data [4]. It estimates the probability that a new set of inputs belongs to every class. The output class is the one that has the highest probability. It also explicitly attempts to model the difference between the classes of data. PCA, in contrast, does not take into account any difference in class. LDA works when the measurements made on independent variables for each observation are continuous quantities. We also make use of LDA's categorical independent variables, in our case the access types (Ethernet / Wifi and Home / Work). LDA is shown in the right illustration of Figure 1.

E. T-distributed stochastic neighbourhood embedding (t-SNE)

t-SNE is a non-linear algorithm for clustering high dimensional data by projecting each measurement point onto a lower dimensional (LD) scatter plot [5]. On a low dimension visualisation, the similarities and differences of the highly dimensional measurement space may allow data exploration and analysis. The LD map is optimised by shifting points using the well known gradient descent, a 1st-order iterative algorithm. The t-SNE algorithm is shown in algorithm 1.

Algorithm 1 Point placement in the LD map.

Data: $X = \{x_1, x_2, ..., x_n\}$
Parameters: $Perp$, T, η, $\alpha(t)$
Result: $Y^{(T)} = \{y_1, y_2, ..., y_n\}$

Compute LD affinities q_{ij}
$p_{ij} \leftarrow \frac{p_{j|i}+p_{i|j}}{2N}$
sample $Y^{(0)} \leftarrow \{y_1, y_2, ..., y_n\}$ from $N(0, 10^{-4}I)$
for for t=1 to T: **do**
    Compute $p_{j|i}$ using data and $Perp$
    Compute gradient $\frac{\partial C}{\partial Y}$
    $Y^{(t)} \leftarrow Y^{(t-1)} + \eta\frac{\partial C}{\partial Y} + \alpha(t)(Y^{(t-1)} - Y^{(\ t-2)})$
end

III. Dataset - fight of the century

A boxing match between Floyd Mayweather (USA) and Manny Pacquiao (Philippines) took place in Las Vegas, May 2015. 10 hours of measurements were taken compromising 40K data points. Three selected endpoints are shown in Table II. `Traceroute` was used as a reachability tool to record

	Philstar	**Showtime**	**Sky**
Domain	philstar.com	sho.com	sky.com
Role	News & entertain. portal	US cable & entertain. network	UK-based & entertain. network
Server location	Arizona, USA.	Amsterdam, NL.	Amsterdam, NL.
CDN provider	Own (on prem)	Akamai	Akamai
Time difference	-7	+1	+1

TABLE II: Three globally accessible streaming sites measured using four nodes on the CheesePi platform.

the number of Internet hops, before and after the event. IP delays were recorded with `ping`, and the front-end server responses with `httping`. It records the round trip times for GET requests to the remote webservers. We also recorded the packet losses as well as the access location and technologies. 'Academic' was a node on the Swedish academic network connected to peers in Europe and the US, it is rarely overloaded[1]. 'Home' is a student residence, prone to occasional overloads, including frequent 802.11 interference.

A. Example - coupled delays

Delays arising from a network and a server interact. A loaded network will result in longer response times from a server, and a busy server will produce longer latency for the network. From an external measurement perspective, these delays might be indistinguishable and change in contribution over time. Systems theory, stability, and coupled systems have a rich history [6]. We will see that the major feature is the delay, and its second moment, variance. in Figure 2 we see how the delays both increase toward the event, the variance

[1] http://stats.sunet.se

#	Feature	Philstar	Showtime	Sky
1.	#hops	12	8	9
2a.	Net. delay	196±36 ms	29±2 ms	2±3 ms
2b.	Serv. delay	418±128 ms	79±130 ms	24±62 ms
3.	Loss	3%	0%	0%
4a.	Acad. over Eth.	418±128 ms	79±130 ms	24±62 ms
4b.	Acad. over WiFi	634±315 ms	154±75 ms	62±456 ms
5a.	Home over Eth.	351±227 ms	75±105 ms	19±86 ms
5b.	Home over WiFi	NA	108±203 ms	48±221 ms
5c.	All active	536±301	245±312 ms	115±250 ms

TABLE III: Rows 1-3 the numerical measured network features, their means and standard deviations, 4-5 the environmental and access technolgies to the sites gievn in Table II.

in the server increases and the systems are indeed *coupled*. Clearly, the average delay is due to the network however, the

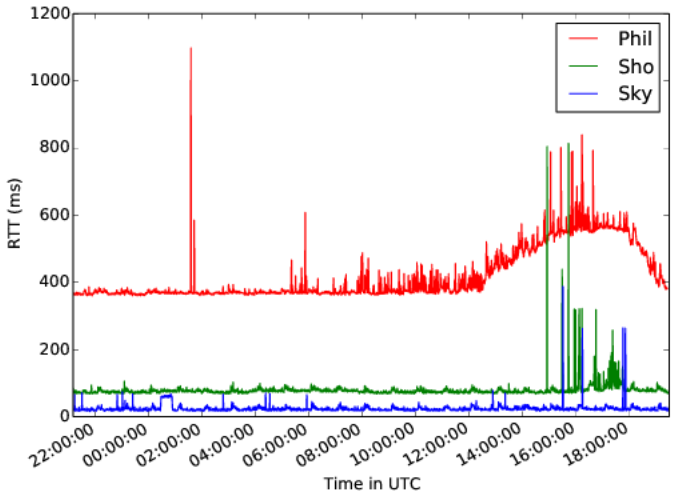

Fig. 2: 24-hour server latency measurements to three streaming sites during the boxing match, measured from a Swedish academic site using Ethernet access.

variance in the delay is due to the server response. To quantify the contributions of the network and server, a scatter biplot of the delay values in the PCA space is shown in Figure 4.

From the whitened, average removed and standard deviation normalised, data, and components overlaid, one can see the contribution of two delay components. Both are positive, showing the server (HTTP) delay being four times that of the network (IP) delay.

IV. Results

A. Visualisation effectiveness

Using dimensionality reduction techniques it is possible to shed light on complex measurement paths and varied environments. Where the values in the covariance matrix are similar (network 'similarity') the points in the visualisation will be close or even overlap. Where there is a significant difference in the measurement space, this should be visible too. Visualisations in this work expose groupings of servers

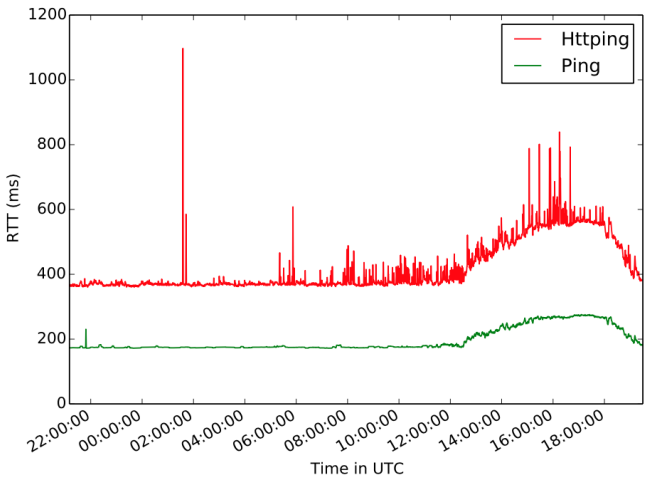

Fig. 3: 24-hour server and network latency measurements to the Philippines server *only*. The server contributes higher delay, but significantly higher variance.

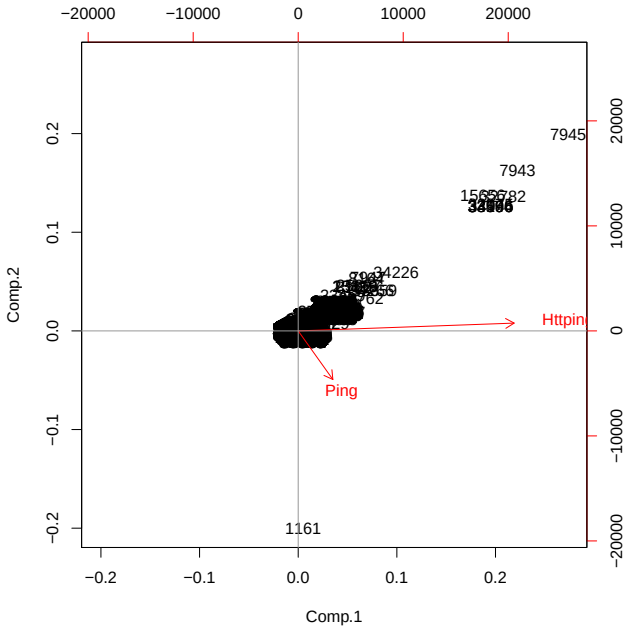

Fig. 4: Scatter biplot of a delay experiment showing the network (Ping) and server (Httping) to a server in the Philippines from Sweden. Individual measurements are in numerals. The lines are the two principal components for each measurement attribute.

for the event, as can be seen in the time series plots. Note, the data is whitened, meaning distributions are shifted and scaled to have zero mean and unit covariance. No information is lost in this process, but it is necessary to compare the techniques presented here. Normalisation is an important step in machine learning.

Figures 5 and 6 show PCA and t-SNE applied to our

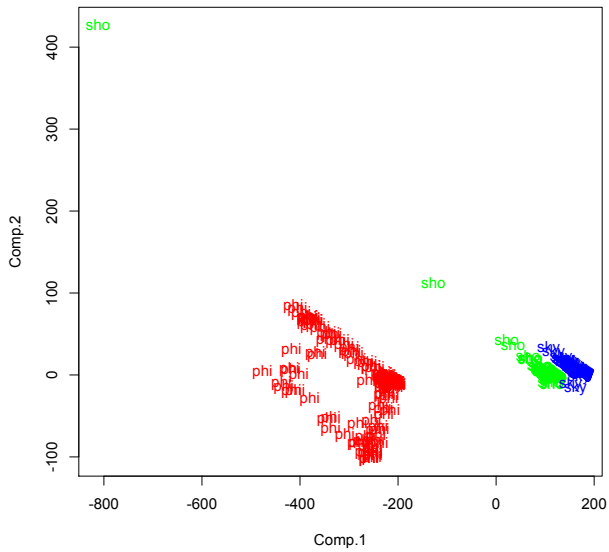

Fig. 5: PCA visualisation of the five-featured boxing measurements (parameterless).

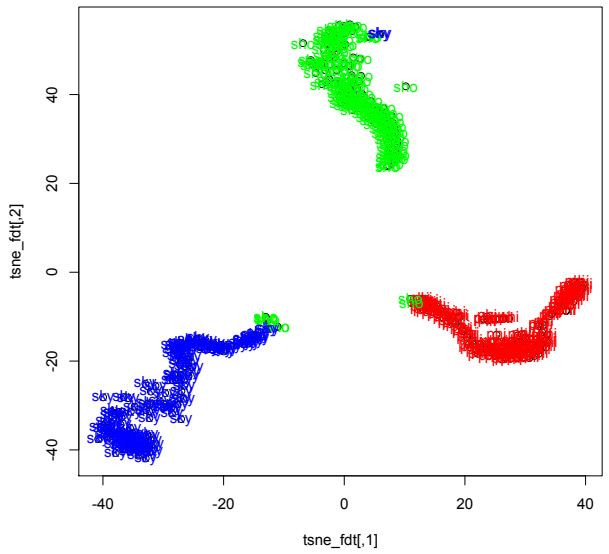

Fig. 6: t-SNE visualisation of the five-featured boxing measurements (using a perplexity of 20).

data, all measurement attributes in Table II were used, and combinations of the environment and technology from Table III. We ran measurements from the environment and technology simultaneously, to correlate the path, location, access technology and measurement environment.

The method of selecting points according to their neighbours, results in clearer separation. A central mass according to the student-t distribution could be identified either by eye or by a clustering method such as Kmeans. Note, at the edges of the groups, there is some bleeding of the measurement points into the groups of the other clusters than they could belong. These are not necessarily errors, especially in the Showtime and Sky cases, but should be looked at in more detail.

B. Runtime performance and memory usage

Performance and memory values are given in Table IV below using R. The values are sensitive to settings such as the kernel chosen in kPCA, grouping and methods in LDA as well as the number of iterations in t-SNE.

Method	Wall clock time (s)	Normalised time (unitless)	Actual memory	Normalised memory (unitless)
Read	0.078	1	233 Kb	1
PCA	0.15	1.9	233 Kb	1
kPCA	55.62	713	21.45 Mb	94
LDA	31.53	404	47.22 Mb	207
t-SNE	44.96	576	56.46 Mb	248

TABLE IV: Normalised and actual execution times and memory usage. Normalisation is with respect to reading the data.

The values in the table do not include the actual plotting, only the computation. The memory use is the size of the data structures holding the 691k measurements[2]. not the allocation and deallocation as the algorithm's progress. This can be substantial, for t-SNE we had to increase the amount of memory using `R_MAX_VSIZE=100Gb` to complete the processing, an alternative is to use GPU processing [7].

V. Related work

We collate most dimension reduction works into three survey articles [8], [9], [10]. The author of the first one is also the co-author of t-SNE. An approach that attempts to address the problems of PCA-like approaches is Sammon mapping [11]. It alters the cost function of classical scaling by dividing the squared error in the representation of each pairwise Euclidean distance by the original Euclidean distance in the HD space. A performance improvement for t-SNE, called Barnes-Hut, has been suggested by its co-creator [5]. Internet measurements by Crovella and Krishnamurthy consider embedding network measurements into high dimensional metric spaces for analysis [12]. Abrahao and Kleinberg looked at the dimensionality properties of the Internet delay space, i.e., the matrix of measured round-trip latencies between Internet hosts using PCA and embedding spaces approaches such as t-SNE [13].

VI. Discussion

1) *Dataset size and the number of features.* Networked measurements often collect many data points. Whereas data processing tools can handle large datasets, high dimensionality mandates method-based approaches as we shone some light on.
2) *Method comparison.* PCA, kPCA and LDA use a Euclidean space for the low dimension visualisation,

[2]691k = 24 hours · 3600 secs · 2 (ICMP and Httping) · 2 access (Eth. and Wifi) · 2 environments (Home and Work)

whereas t-SNE uses a non-Euclidean space for the low dimension visualisation. PCA cannot directly deal with labelled data, or needs extra methods to deal with labels. PCA and kPCA are unsupervised and cannot use class information. Whereas LDA and t-SNE are supervised techniques and can use class information, they provide better visualisation at higher computation costs.

3) *Model performance.* Vanilla PCA is 70-580 times faster than the other methods and requires 94-228 times less memory. kPCA with its non-linear mappings can use significant memory, but saves on cycles with the kernel 'trick'. LDA and t-SNE are CPU and memory demanding. Classification is usually not needed with t-SNE, but experimentation with its Perplexity parameter almost is. For future work, this should be explored further.
4) *Parameterisation.* PCA is parameter-free and thus has large advantage other the other methods. kPCA needs a kernel parameter (e.g. Gaussian) which not only effects the running time and visualisation, can actually misleading results. With complex data, kPCA requires substantional exploration as detailed in this excellent paper [14]. LDA requres the number of components as a parameter (two as shown in Figure 1). t-SNE requires at least one parameter, but also requires a number of execution options, significantly effecting the running time and visualisation appearance. An example is the number of layout iterations shown. Since visualisation is subjective, t-SNE allows for a number and the user to select the most 'attractive'. In this work we perfomed only one iteration to be as fair as possible to the other methods. In reality, a number should be selected.
5) *Visualisation effectiveness.* PCA, k-PCA ad LDA produce good visualisations, t-SNE probably the best overall (see our web site). Perplexity can range between 5-50, where we found 20 to be acceptable for our data. Visualisation aids help users how to utilise t-SNE.

VII. Conclusions

We have motivated, explained and compared four dimensionality reduction techniques for the analysis of network data in this relatively short paper. Such techniques have thus far been quite overlooked in the networking community by resorting to multiple 2D plots or summary statistics. Visualisation in other communities, however, is very commonplace and useful.

The 4 techniques presented preserve all the measurement correlations, but project them down into easier comprehend two dimensions. The techniques even help resolve complex dependencies, as shown in the example of coupled delays. Generally, we found parameterless PCA to perform well, but it is worthwhile experimenting with the others, especially if one has categorical data.

References

[1] I. T. Jolliffe, *Principal Component Analysis*. Berlin; New York: Springer-Verlag, 1986.

[2] L. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[3] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Phil. Mag.*, vol. 6, no. 2, pp. 559–572, 1901.

[4] R. A. Fisher, "The precision of discriminant functions," *Annals of Eugenics*, vol. 10, no. 1, pp. 422–429, 1940. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1940.tb02264.x

[5] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, Jan. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2697068

[6] L. M. Pecora and T. L. Carroll, "Master stability functions for synchronized coupled systems," *Phys. Rev. Lett.*, vol. 80, pp. 2109–2112, Mar 1998. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.80.2109

[7] D. M. Chan, R. Rao, F. Huang, and J. F. Canny, "t-SNE-CUDA: GPU-Accelerated t-SNE and its Applications to Modern Data," *CoRR*, vol. abs/1807.11824, 2018. [Online]. Available: http://arxiv.org/abs/1807.11824

[8] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," 2014.

[9] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *Journal of Machine Learning Research*, vol. 16, no. 89, pp. 2859–2900, 2015. [Online]. Available: http://jmlr.org/papers/v16/cunningham15a.html

[10] L. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review," 2008.

[11] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on computers*, vol. 18, no. 5, pp. 401–409, 1969.

[12] M. E. Crovella and B. Krishnamurthy, *Internet measurement : infrastructure, traffic, and applications*. Hoboken, NJ: J. Wiley and sons, 2006. [Online]. Available: http://opac.inria.fr/record=b1119782

[13] B. Abrahao and R. Kleinberg, "On the internet delay space dimensionality," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 157–168. [Online]. Available: https://doi.org/10.1145/1452520.1452541

[14] M. A. Alam and K. Fukumizu, "Hyperparameter selection in kernel principal component analysis," *Journal of Computer Science*, vol. 10, pp. 1139–1150, 01 2014.