Uncovering interpretable relationships in high-dimensional scientific data through function preserving projections

IOP Publishing Mach. Learn.: Sci. Technol. 1 (2020) 045016 https://doi.org/10.1088/2632-2153/abab60

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
19 March 2020

REVISED
18 July 2020

ACCEPTED FOR PUBLICATION
31 July 2020

PUBLISHED
15 October 2020

PAPER

Uncovering interpretable relationships in high-dimensional scientific data through function preserving projections

Shusen Liu, Rushil Anirudh, Jayaraman J Thiagarajan and Peer-Timo Bremer

Center for Applied Scientific Computing (CASC), Computing Directorate Lawrence Livermore National Laboratory 7000 East Ave, Livermore, CA 94550, United States of America

E-mail: liu42@llnl.gov, anirudh1@llnl.gov, jayaramanthi1@llnl.gov and bremer5@llnl.gov

Keywords: exploratory data analysis, dimensionality reduction, scientific data, linear projection

Abstract

In many fields of science and engineering, we frequently encounter experiments or simulations datasets that describe the behavior of complex systems and uncovering human interpretable patterns between their inputs and outputs via exploratory data analysis is essential for building intuition and facilitating discovery. Often, we resort to 2D embeddings for examining these high-dimensional relationships (e.g. dimensionality reduction). However, most existing embedding methods treat the dimensions as coordinates for samples in a high-dimensional space, which fail to capture the potential functional relationships, and the few methods that do take function into consideration either only focus on linear patterns or produce non-linear embeddings that are hard to interpret. To address these challenges, we proposed function preserving projections (FPP), which construct 2D linear embeddings optimized to reveal interpretable yet potentially non-linear patterns between the domain and the range of a high-dimensional function. The intuition here is that humans are good at understanding potentially non-linear patterns in 2D but unable to interpret non-linear mapping from high-dimensional space to 2D. Therefore, we should restrict the projection to linear but not the pattern we are seeking. Using FPP on real-world datasets, one can obtain fundamentally new insights about high-dimensional relationships in extremely large datasets that could not be processed with existing dimension reduction methods.

1. Introduction

The rapid advances in both experimental and computational capabilities have resulted in a deluge of data being collected from either computer simulations or experiments. To gain insights into the underlying phenomena, it is imperative to uncover patterns and relationships in the resulting high-dimensional (HD) data. Especially in the initial exploratory phase of the analysis (*i.e.* exploratory data analysis (Tukey 1977)), visualization techniques are often essential to provide a holistic understanding of the data and help develop an intuition for the domain scientists. Since most common HD data encoding techniques like scatterplot matrices (Carr *et al* 1987) and parallel coordinates (Inselberg and Dimsdale 1990) do not scale gracefully to either many dimensions or large sample sizes, dimension reduction techniques are typically the method of choice. In this context, one then has to trade off the complexity of the embedding with the interpretability of the results. For example, linear dimension reductions, such as principal component analysis (PCA) (Jolliffe 2011), are easy to interpret yet do not generalize to complex patterns, whereas non-linear techniques like t-stochastic neighborhood embedding (t-SNE) (van der Maaten and Hinton 2008) are difficult to interpret. Furthermore, these approaches are generally designed to preserve as much of the geometric structure of HD data as possible. However, this structure itself often is not of primary interest, or in the case of space-filling experimental designs, no meaningful structure exists. Instead, one is interested in understanding the behavior of one or more *response* functions with respect to the input data. Here, response function(s) refer to the main quantities of interest defined at each data sample, e.g. diagnostic measurements from an experiment or simulation outputs. Consequently, one needs an approach sensitive to the response functions rather than using a generic dimension reduction in hopes that it may explain some aspect of the response.

Additionally, the technique should scale to large data, high dimensions, and non-linear relationships, while remaining interpretable and reliable.

Function preserving projections (FPP) achieve these objectives by deriving optimal linear projections with respect to user-selected response functions. More specifically, rather than aiming to preserve the entire neighborhood structure of high-dimensional data in hopes of finding interesting patterns, we deliberately search for the best linear projection, such that the chosen response functions create an interpretable pattern in the projected space. An important consequence of focusing on the projected response function is that FPP inherently ignores domain variables and structures not pertinent to the response and does not require the domain to have a low intrinsic dimension. The latter is a key property in many applications, most notably simulation ensembles, where the domain is defined as a uniform sampling of a high dimensional hypercube. In these cases, dimension reduction is futile as—by definition—there exists no low dimensional structure in the domain one could discover and thus many of the traditional techniques do not apply. Compared to supervised dimensionality reduction techniques that either produce non-linear embedding (Nonato and Aupetit 2018), or focus solely on linear relationships (Chin *et al* 1998, Li 1991), the proposed method is fundamentally different as it focuses on producing linear projection that can uncover interpretable non-linear patterns of the function.

The interpretable aspect of the proposed method is achieved through the linear projection constrain and the explicitly designed 2D optimization cost. We consider a projected pattern interpretable if it can be roughly approximated by a chosen regressor and by adjusting the type and order of the regressor, *i.e.* polynomial vs. exponential, linear vs. non-linear, etc we allow users to choose the complexity of the pattern they deem acceptable, and by forcing the projected function to be explained using a simple and easily understandable regressor (e.g. low-order polynomial models), combined with the fact that the projection is linear, we can provide valuable and at the same time understandable insights that are otherwise hard to tease out. The key intuition driving the development of FPP is the fact that interpreting non-linear transformation is challenging yet humans are highly skilled in recognizing and understanding simple non-linear patterns in 2D. By restricting the initial projection to a linear map, FPP preserves the interpretability of the resulting plots, *i.e.* axis labels, while non-linear regressors enable us to discover noisy and non-linear relationships.

Conceptually, FPP is a dual approach to kernel machines in machine learning, which employ non-linear, and often infinite-dimensional mappings to enable the use of simple linear models for complex data. From a visual exploration standpoint, we argue that the use of linear models in 2D is not necessary since humans can still interpret more complex relationships. On the other hand, non-linear mappings of the data coordinates are not explainable, thereby making the subsequent analysis also highly opaque. Another crucial feature of FPP is that it supports much larger data than related techniques and enables unified analysis with multiple response functions, wherein a single 2D projection is identified that jointly preserves all responses. Finally, an often-overlooked challenge for view-finding or any pattern detection algorithm is that, in higher dimensions, it is challenging to qualitatively distinguish between meaningful structure and artifacts. This behavior can be directly attributed to the curse of dimensionality, where the sample sizes used are not sufficient to make meaningful inferences about the data. In the case of view-finding approaches like FPP, this manifests as overfitting, where one can almost always create a seemingly meaningful pattern given low enough sample counts and sufficiently high dimensions. To address this problem we augment the 2D embeddings with the equivalent of a p-statistic that quantifies the likelihood of the given hypothesis (the observed pattern) to occur in a random function. This, for the first time, provides a quantitative and easy to interpret indicator on how reliable a given visualization is likely to be, which is crucial to confidently infer new insights. Using several case studies we show that FPP provides comprehensive insights that cannot be easily obtained using conventional techniques.

Our key contributions are itemized below:

- Introduce a linear projection method for capturing human-interpretable pattern of a high-dimensional function;
- Demonstrate the flexibility and scalability of the proposed method for capturing patterns in both continuous scalar functions as well as class labels defined on high-dimensional domains with up to million of samples and tens of thousands of dimensions; and
- Introduce a *p-value* like evaluation scheme for captured visual patterns through the lens of hypothesis testing.

2. Related work

Even though FPP produces linear projections, it is fundamentally different from existing linear dimension reduction techniques like principal component analysis (PCA) (Jolliffe 2011). Dimension reduction, as

currently understood, is typically aimed at preserving global (PCA) or local neighborhood structures (locality preserving projection (Xiaofei and Niyogi 2004)) of the high-dimensional point geometry. The resulting projection is then colored according to a response function in hopes of highlighting interesting relationships. However, when analyzing particular response functions the complete geometry of the point set may not be relevant and can even be detrimental by introducing 'spurious' variations unrelated to the response of interest. Instead, FPP directly targets the response functions of interest and preserves only the aspects of the high dimensional geometry relevant to the problem.

In this context, FPP is similar to cross decomposition approaches, such as canonical component analysis (CCA) (Hardoon *et al* 2004) and partial least square (PLS) (Chin *et al* 1998). CCA aims to find the subspace that best aligns the domain and the range of a high-dimensional function. However, this produces a subspace that is at most equal to the minimal dimension of either the domain and the range. Consequently, for scalar functions, one can only find a 1D subspace, rather than a 2D projection, and for more than two response functions a secondary projection must be added for visualization which results in a suboptimal projection. Partial least square (Chin *et al* 1998) does not have this limitation but is restricted to a linear regressor making it difficult to identify even simple non-linear patterns, *i.e.* a circle (see section 4). Similarly, inverse sliced regression (Li 1991), which utilize an inverse regression formulation to reduce the dimension of the input with respect to the response (function), is also limited to linear correlation structure.

For visualization of the non-linear structure of the high-dimensional points, many non-linear dimensionality reduction techniques (van der Maaten and Hinton 2008, Kruskal 1964) have been proposed. However, while they can capture some intrinsic structure of a point sample it is difficult to connect the observed patterns back to the domain as the axes no longer have well-defined meaning and distances can be heavily distorted. Furthermore, similar to linear projections, these techniques do not consider the response function in creating the projection. As a result, a projection that explains the response well is a fortunate coincident rather than a deliberate design.

The use of dimensionality reduction for visualization is a well explored subject (Nonato and Aupetit 2018, Espadoto *et al* 2019). As discussed in Nonato and Aupetit (2018), these techniques can be extended to interpret values defined on a high-dimensional domain by incorporating supervised information, for example, through altering the distance representation among samples, e.g. encode class information by artificially reducing intra-class distances. However, these approaches are different from the proposed methods in two significant ways: First, FPP deliberately considers only the pattern of the function, whereas semi-supervised methods often try to capture both the structure of the samples as well as the function on these samples. Second, the explicit computation of inter-sample distance in both high-dimensional and low-dimensional space is what we aim to avoid, as it is often the computational bottleneck for a scalable solution. By directly focusing on the pattern of function in the low-dimensional space, we are able to scale to extremely large datasets with very limited computational resources that would be impossible to process using inter-sample distances. Moreover, since we focus on introducing a novel linear projection technique instead of povide methods to interpret existing onces, the proposed methods is also fundamentally different from the visualization techniques designed for explaining dimensionality reduction results (Liu *et al* 2014, Chatzimparmpas *et al* 2020).

The technique most similar to FPP is projection pursuit regression (PPR) (Friedman and Stuetzle 1981), which has been designed as a universal high-dimensional function approximator for regression tasks. The PPR fits a linear combination of multiple 1D non-linear transformations of linear combinations of variables in the data. The non-linear mapping allows PPR to capture certain non-linear patterns. Similar to FPP, the PPR formulation can also be considered as a dual of the kernel regression as explored in Donoho and Johnstone (1989). However, designed as a function approximator, the performance of PPR improves as we increase the number of 1D non-linear transformation components, which can lead to challenges in its interpretation. With FPP, we instead directly fit a non-linear model in the 2D projected space, which not only allows intuitive visualization but also simplifies the optimization process which allows us to efficiently scale the proposed technique to extremely large sample size and dimensions.

3. Method

As discussed in the introduction, the growing need for exploring large and complex high-dimensional dataset calls for visualization tools that 1) produce interpretable embedding; 2) are capable for capturing non-linear pattern; 3) and scale to large sample sizes and many dimensions. For interpretability, contrary to many high-dimensional data visualization approach (e.g. t-SNE or MDS) that employ a complex map from high-dimensional to 2D space, we focus on a simple linear transformation that produces a 2D embedding with well-defined axes.

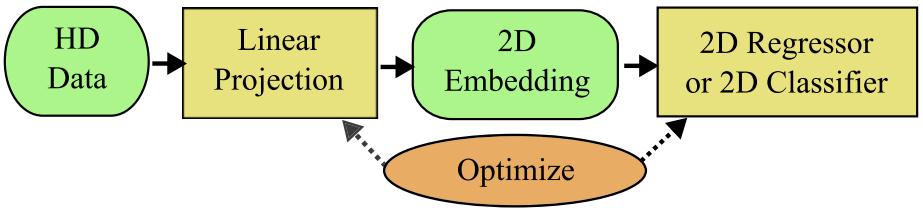


Figure 1. Overview of the computation pipeline for the proposed function preserving projection approach. We jointly optimize the linear projection and the 2D regressor (or classifier) for identifying human-interpretable pattern in high-dimensional functions.

3.1. Function preserving projection

To capture the potentially non-linear structure of the function, we frame the pattern discovery as a non-linear 2D regression problem, where the choice of the regressor, *i.e.* the polynomial degree, provides direct control over the visual complexity a user expects or is willing to consider as salient structure. In the most basic form, we can consider the problem as a joint optimization of both dimensionality reduction and regression (see figure 1) that can be formulated as follow:

For a given HD dataset of N samples in D dimensions, $\mathbf{X} \subset \mathbb{X}$, FPP infers d-dimensional embeddings $\mathbf{Y} \subset \mathbb{Y}$, based on a response function f defined at each data point $f_i \in \mathbb{F}, \forall i = 1 \cdots N$. Here, $\mathbb{X}$ and $\mathbb{Y}$ denote the input and the embedded spaces, respectively. The response function space is defined as either $\mathbb{F} \subseteq \mathbb{R}$, in case of continuous response functions, or as $\mathbb{F} \subseteq \Omega$ when f assumes one of K discrete values. This leads to the following general formulation of FPP:

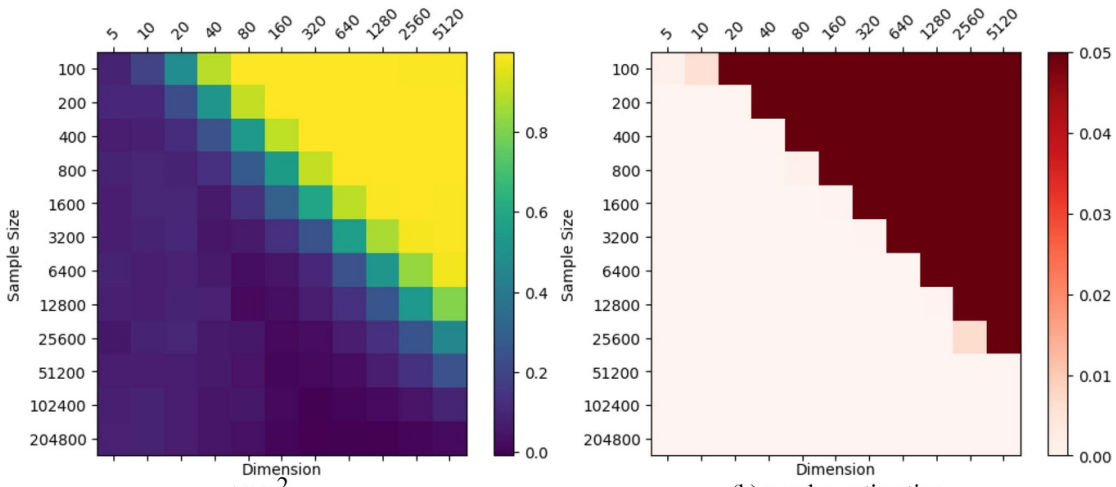
$$\underset{\mathbf{P},\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{S}\left[f_i, g(\mathbf{y}_i;\theta)\right], \quad \text{where } \mathbf{y}_i = \mathbf{P}^T \mathbf{x}_i. \tag{1}$$

In this formulation, $g : \mathbb{Y} \mapsto \mathbb{F}$ denotes the mapping function, with parameters θ, between the embedded space and the response function, $\mathbf{P} \in \mathbb{R}^{D \times d}$ is a linear orthonormal projection applied to each data sample $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathcal{S}$ is a scoring function used to evaluate the quality of mapping g. In order to achieve interpretability, FPP relies on linear projections $\mathbf{P}$ and for visualization purposes d is typically fixed at 2. One can then capture non-linear relationships by allowing sufficient flexibility for the mapping function g, *i.e.* using higher order regression models. For continuous f, the examples below use polynomial regressors, where the polynomial degree directly controls the complexity of the inferred structure. However, other regressors could easily be integrated as well. As scoring function $\mathcal{S}$ any one of the standard goodness of fit measure can be used, such as the mean squared error (MSE). In the case of classification when f is discrete, g is defined as a softmax classifier to predict the K class labels. In these cases $\mathcal{S}$ is defined as the cross entropy between true and predicted response values. Finally, equation (1) can be extended to multiple response functions $f^l, l = 1 \cdots L$ as follows:

$$\underset{\mathbf{P},\{\theta^l\}}{\operatorname{argmin}} \frac{1}{LN} \sum_{l=1}^{L} \sum_{i=1}^{N} \mathcal{S}\left[f_i^l, g(\mathbf{y}_i;\theta^l)\right], \quad \text{where } \mathbf{y}_i = \mathbf{P}^T \mathbf{x}_i. \tag{2}$$

Here, our goal is to infer an unified projection $\mathbf{P}$ that simultaneously recovers the L response functions.

To solve equation (1) requires incorporating the constraint $\mathbf{P}^T\mathbf{P} = \mathbb{I}$ to ensure that the columns of $\mathbf{P}$ are orthonormal and the projection constructs a valid linear subspace. More specifically, FPP leverages the popular deep learning framework *TensorFlow* (Abadi *et al* 2016) to implement a projected gradient descent (PGD) optimization. The linear projection is realized with a dense layer with weight matrix of size $\mathbb{R}^{D \times d}$, and the orthonormality constraint is enforced by projecting estimated weights onto the Stiefel manifold, the set of all orthonormal matrices of the form $\mathbb{R}^{D \times d}$, through singular value decomposition (SVD) (Golub and Reinsch 1971). The main consideration to utilize SVD is due to the excellent support through deep learning frameworks like Pytorch and Tensorflow that provide efficient and differentiable implementations for SVD. As a result, they yield far superior performance and stability than a hand-coded Gram-Schmidt orthonormalization procedure. However, FPP is agnostic to the choice of orthogonalization and alternative approaches like QR decomposition will also work—as long as the operations are differentiable for *autograd* computation. The embeddings from the linear projection, $\mathbf{y} = \mathbf{P}^T\mathbf{x}$, are then used for predicting the response f using a non-linear mapping g. The detailed steps of FPP are summarized in Algorithm 1. Note that for small d, *i.e.* $d = 2$ the SVD step is computationally efficient and consequently FPP scales to tens of millions of



(a) R^2 score (b) p-value estimation

Figure 2. Here we investigate the trustworthiness of the captured pattern in respect to different dimension and sample size combinations. In this example, we focus on the setup utilizing a 2D polynomial regressor (degree 4). By examining the potential patterns in totally random data, we can estimate how likely the observed pattern is spurious correlation rather than salient structure. In (a), we show the average R^2 score of the fitted regressors on random datasets at different conditions. We can then utilize the R^2 score samples on random datasets to estimate a p-value for a given data and its corresponding projection. In (b), we show the p-value assuming a projection gives an R^2 score of 0.5 for each of the dimension and sample size combinations.

samples and tens of thousands of dimensions. Our implementation can be found at https://github.com/LLNL/fpp.

Algorithm 1: Function Preserving Projections.

Input: Domain $\mathbf{X} \in \mathbb{R}^{D\times N}$ and response function $f \in \mathbb{R}^N$; Scoring function $\mathcal{S}$; Learning rate γ, mini-batch size b
Output: Projection matrix: $\hat{\mathbf{P}} \in \mathbb{R}^{2\times D}$; Parameters $\hat{\theta}$ for g
Initialize Randomly initialize $\hat{\theta}$ and $\hat{\mathbf{P}}$ (orthonormal matrix)
while exist mini-batch $\widetilde{\mathbf{X}}, \widetilde{f}$ from $\mathbf{X}$, **fdo**
  // `project input HD data onto 2D`
  $\widetilde{\mathbf{y}}_i \leftarrow \hat{\mathbf{P}}^T \widetilde{\mathbf{x}}_i, \forall i = 1 \cdots b$;
  // `predict response function and compute goodness of fit`
  $L \leftarrow \sum_{i=1}^{b} \mathcal{S}[\widetilde{f}_i, g(\widetilde{\mathbf{y}}_i; \hat{\theta})]$;
  // `update parameters`
  $\hat{\theta} \leftarrow \hat{\theta} - \gamma \nabla_\theta(L)$;
  $\hat{\mathbf{P}} \leftarrow \hat{\mathbf{P}} - \gamma \nabla_{\mathbf{P}}(L)$;
  // `enforce orthonormality constraint`
  $\mathbf{U}, \Sigma, \mathbf{V^T} \leftarrow \mathrm{SVD}(\hat{\mathbf{P}})$;
  $\hat{\mathbf{P}} \leftarrow \mathbf{U}$;
end
return $\hat{\mathbf{P}}$

3.2. Evaluation schemes for visual patterns

For any pattern-finding scheme, it is imperative to evaluate the trustworthiness of the identified pattern. FPP provides an end-to-end solution that consists of both a linear projection and a 2D regression/classification model. So, it can and will overfit to the datasets if there is a large number of dimensions, *i.e.* a large number of parameters in the projection matrix, with limited number of samples. We address this challenge from two perspectives: First, like all statistical inference problems, we can split the data into training and testing set, fit the projection using training data, and compare the results on both the training and testing set. If the identified pattern is due to a salient correlation we expect both projections to result in a very similar structure. Alternatively, we can consider the problem as a hypothesis test, by defining a confidence value (analog to a *p-value*), which describes how likely a pattern of similar strength can be found in random data. However, different from the standard way of computing *p-value* for linear regressor, we measure such a probability by repeatedly generating randomness in the input data through shuffling of function values and then compute FPP on these randomized inputs.

Such a test can also provide us with general guidelines on whether we should be concerned about potential overfitting at a given sample count and data dimension. As illustrated in figure 2(a), for the given

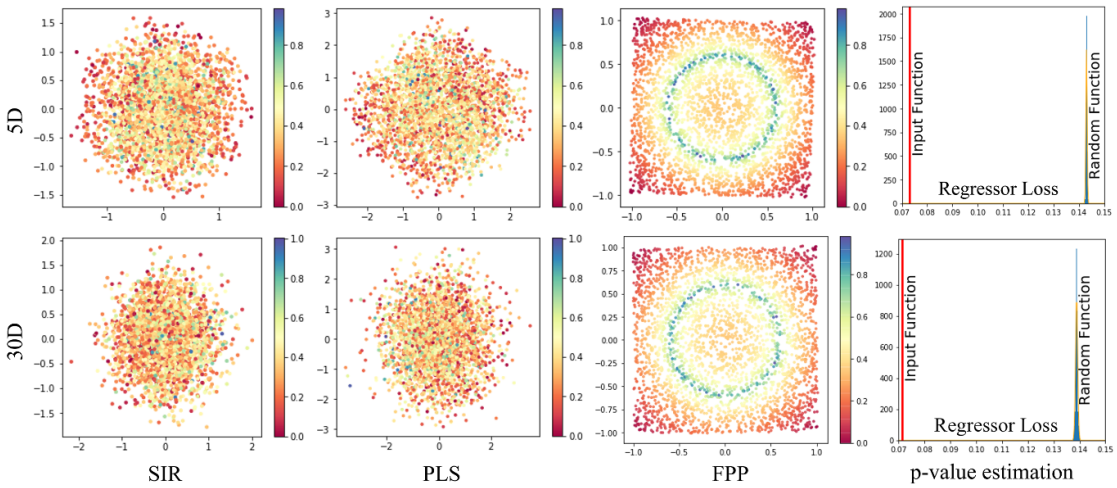


Figure 3. Synthetic data. The top and bottom rows are results from a 5D and 30D function, respectively. Here we show that the function preserving projection (FPP) reliably identifies the circle structure in both the 5 and 30-dimensional dataset, whereas the partial least square (PLS) and sliced inverse regression (SIR) fail. The rightmost column illustrates the p-value estimation scheme for the observed pattern in the projections. Here, we show the distribution of the regression loss values of fitted models on randomized data and then compared it against the loss value obtained from the original function. In the plot, the x-axis is the regression loss value. The regression loss for the given function is highlighted by the red line, where as the loss value distribution on the randomized data is shown in blue. The separation between them indicate that the patterns we observed in the projection are statistically significant.

2D polynomial (degree 4) regressor, we show its R^2 score when fitted to randomly generated data of different dimension and size. As expected, overfitting is more likely to happen as the dimension increases or as sample count decreases. Subsequently, we utilize the R^2 score samples on random datasets to estimate a p-value for a given projection. As illustrated in figure 2(b), we show p-values assuming the projection give a R^2 score of 0.5 for configurations. The colormap is clamped at $p = 0.05$, which reveals a clear line separating the significant and non-significant sides. Such an observation indicates that a constant factor exists between sample size and dimension size for finding a trustworthy pattern of the regression problem (in this case, the sample size should be at least ten times larger). For a more general setup, we can also obtain the p-value estimation from the training/evaluation loss of FPP (instead of R^2), which can be used for both the regression and classification scenarios. In case of the regression setup, we adopt a mean squared error (MSE) loss, and for the classification case utilize a cross-entropy loss.

Finally, the proposed framework is not limited to any particular 2D regressor or classifier, provided they are differentiable in order to utilize the existing SGD optimization framework. Polynomial regressors are effective for capturing the non-linear and low-frequency pattern human can easily comprehend, but other regressors likely will work as well. One thing to note is that different types of regressors will impose different priors on the patterns. Therefore, we can utilize the selection of regressor and their setup, *i.e.* the degree of the polynomial regressor, as a tunable knob in the system to focus on different types of pattern or complexity. However, we should expect to see potentially different overfitting behaviors due to the choice of the model. For certain datasets with high extrinsic but low intrinsic dimension, one can also utilize random projections as a pre-process to lower the dimensionality of the problem. By its nature, random projections are highly unlikely to cause overfitting but the resulting lower dimensionality significantly reduces the overfitting risk for the subsequent FPP projection.

4. Results

In this section, we demonstrate the applicability of the proposed method to synthetic data as well as to dataset from real-word applications for both regression and classification problems.

4.1. Synthetic dataset

As a synthetic experiment, we define a scalar function on a uniformly sampled high-dimensional domain. Here, the function has a circular pattern (see the third column of the figure 3) in a 2D subspace of a 5D and 30D domain, respectively. The synthetic data is generated with the following steps: (1) generate uniform random samples in 2D (between -1.0 to 1.0 for each dimension); (2) assign the function pattern by computing the distances from samples to a circle in 2D; (3) populate uniform random values (between -1.0 to 1.0) to the rest of the dimensions to lift the original 2D domain into a higher-dimensional space, in which

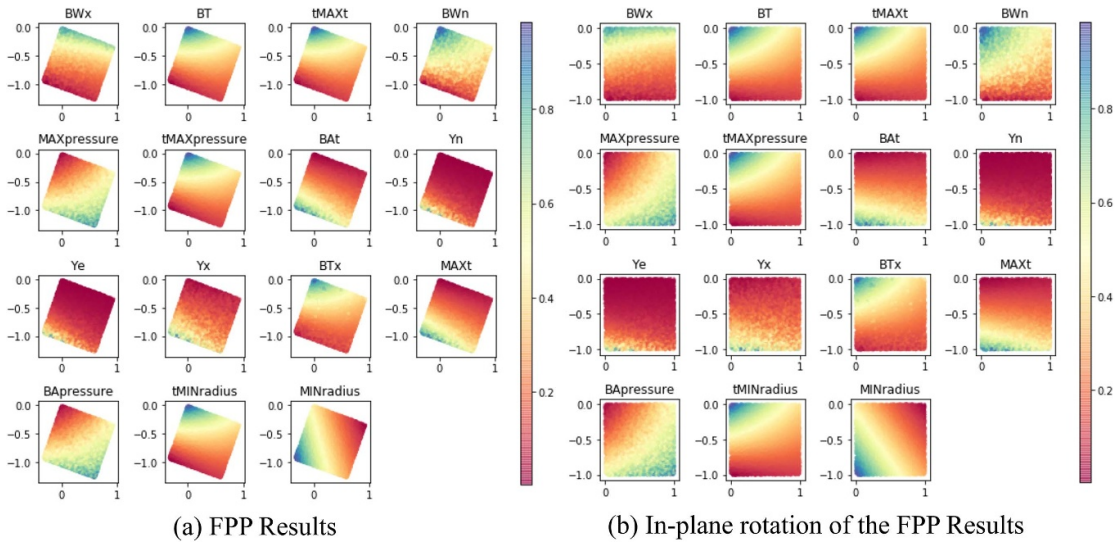


Figure 4. A demonstration of multi-function projection capability on a physical simulation ensemble dataset. In this ensemble dataset, we have 5 input parameter and 15 output scalar functions. Here, we aim to identify a single 2D embedding, in which all functions' variation can be explained. The plot on the left shows the subspace identified by FPP, in which all subplots have the same embedding configuration and colored by the values of the 15 outputs, respectively. The configuration can be simplified by making an in-plane rotation, as shown on the right plot. The two directions in these plots correspond to two of the input parameters.

the sample are uniformly sample in each of the dimensions; (4) apply random rotation to high-dimensional points, so the pattern of interest is not in an axis-aligned subspace. The last step is to ensure we can uncover the pattern in an arbitrary linear subspace.

The top row of figure 3 shows projections from the 5D dataset and the bottom row are from the 30D dataset. Due to the linear assumption, both partial least square (PLS) and sliced inverse regression (SIR) fail to capture the circular pattern. By focusing on the visual domain and relying on a non-linear regressor (in this case, a polynomial regressor of degree 3), the proposed FPP approach can easily reveal the pattern in both the 5D and 30D domain. On the rightmost column, we illustrate the significance estimation (p-value) of the pattern captured by the proposed technique. The blue histogram shows the regression loss value distribution for a sample of 300 randomized functions (*i.e.* a random shuffle of the function values), whereas the red line marker indicates the loss value for the input function. This plot provides insights on whether we overfit to the data by illustrating how likely we will be able to find a pattern of similar strength in the random function. The clear separation between the randomized function regression loss distribution and the input function loss distribution lead to an estimated of p-value 0.0, which indicates a strong evidence against the hypothesis that the observed pattern is from spurious correlation.

4.2. Application to multi-variate physical simulation dataset

In the synthetic dataset, we use FPP to produce a 2D embedding based on one function of interests. However, in many applications, we are interested in the joint behavior of multiple scalar function, *i.e.* output properties of simulation ensembles. As discussed in section 3, we can easily extend the single function formulation to multiple ones. In the following example, we give an example of a multi-function projection, where we produce a single 2D embedding that explains all major variation of the functions. The application is a physical simulation ensemble (1 M samples) produced by a recently proposed semi-analytic simulation model (Gaffney *et al* 2014, Springer *et al* 2013) for inertial confinement fusion (ICF) (Gaffney *et al* 2020). The simulator has a 5-dimensional input parameter space and produces several images of the implosion as well as 15 diagnostic scalar outputs. Among the five input parameters, three of them describe the shape of the implosion, whereas the rest two are aggregated parameter that represents other conditions of the experiment. The 15 scalars capture physical measures pertinent to analyze various aspects of the ICF process, including energies in different spectra, pressures, etc. For this example, we want to understand what are the main driving factors for not one but all 15 scalar outputs.

We can explore such a relationship by producing a simple 2D projection that would best capture the changes and pattern of the 15 scalars. As illustrated in figure 4(a), by utilizing a degree-3 polynomial regressor for each function, we identify a single 2D projection (colored by 15 different scalar values with a shared color bar and all output scalars are normalized to 0-1) that explains most of the variation of all scalar

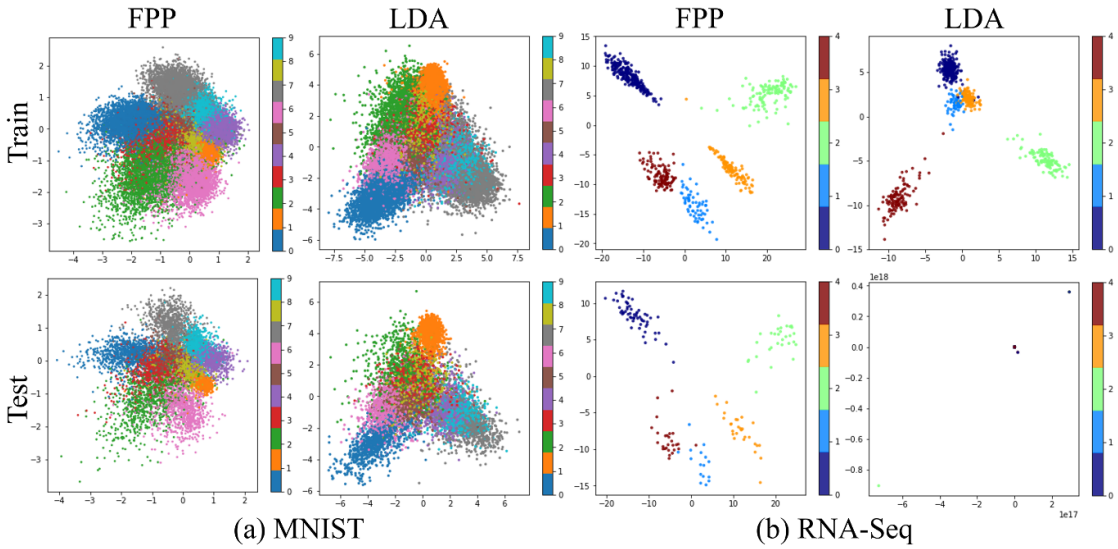


Figure 5. FPP projection examples where the function is the class labels. In (a), the MNIST image data with 10 labels is illustrated, whereas the RNA sequence (RNA-seq) data with 5 labels is shown in (b). The proposed method generates more clearly separation for each class for both data. In particular, the proposed method also produces a more robust projection for the RNA-seq example, where the LDA is overfitted to the training data and fails to generalize to test set.

outputs in the simulation ensemble. As it returns out, as shown in figure 4(b), if we apply in-plane rotation, the two dominating directions corresponds to two of the input parameters. According to the physicists, the other three parameters (out of five input parameters) are shape parameters, therefore, they will mostly impact the generated image instead of the scalars. For example, *MAXpressure* is equally influenced by both of the non-shape parameters. This example provides a real word verification of the proposed ability to capture a shared configuration, which helps interpret a set of functions defined in the same domain. The fitted model has a loss of 0.037 4. To estimate the p-value, we obtain the mean and variance of the loss distribution on randomized function computed from 300 samples, which lead to an estimated p-value of 0.0.

4.3. Application to labeled datasets

In previous examples, we have demonstrated the effectiveness of FPP for projecting continuous, high-dimensional functions, *i.e.* for regression problems. For classification problems, *i.e.* find the 2D projection that best separate samples with different labels, we can simply replace the 2D regressor with a 2D classifier. Here, we utilize a 2D logistic regression classifier (with an additional non-linear layer) to drive the selection of the linear projection. In figure 5(a), we compare the proposed FPP with linear discriminant analysis (LDA) (Fisher 1936) on the MNIST dataset [1] that consists of 60 K sample of 28 by 28 grey scaled images of handwritten digits from 0-9. We can see that FPP finds a 2D linear projection that separates different digits' images better than the LDA result. We also apply the method to high-dimensional RNA sequence data [2], which has more than 20 531 feature dimensions yet only 801 samples. Due to the extremely high-dimensional and the very low sample count, there is a high potential for overfitting. As discussed in previous examples, we can obtain the p-value 2.027e−07, which give high confidence on the captured structure. Alternatively, we can validate the trustworthiness of both the FPP and LDA results by split the datasets into training and test and then evaluate the trained models on the test set. As shown in figure 5(b), not only does FPP separate the class better than LDA, but it also produces a more robust projection that generalizes well to the test set, unlike the LDA projection which entirely fails on the test set. When examining the LDA projection matrix, we notice only 3 of the 20 K dimensions are contributing to the projection, which leads to a degenerate projection for the test set. The reason LDA produces worse results than FPP is due to the fundamental difference in their formulations. Firstly, LDA captures linear separability and assumes each class has a Gaussian distribution, which can be restrictive, e.g. such assumptions lead to more tendency for the LDA to overfit to the training dataset, result in a very sparse and less generalizable model. For the proposed FPP, we not only did not assume the distribution of each class but also did not require linear separability in 2D (by adding a hidden neural network layer before the cross-entropy computation), which led to better separation in terms of human perception for a linear projection.

[1] http://yann.lecun.com/exdb/mnist/.

[2] https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq.

Table 1. Quantitative Comparison with Linear Discriminant Analysis.

Dataset	Silhouette	Davies-Bouldin	Homoge-neity	Complete-ness	Mutual-Information	Timing(s)
MNIST train FPP	**0.139**	**1.589**	**0.575**	**0.574**	**0.575**	17.7
MNIST train LDA	0.069 8	6.339	0.503	0.480	0.491	**9.93**
MNIST test FPP	**0.138**	**1.595**	**0.571**	**0.570**	**0.570**	N/A
MNIST test LDA	0.063 5	4.642	0.487	0.478	0.483	N/A
RNA-Seq train FPP	**0.781**	**0.307**	**1.0**	**1.0**	**1.0**	**1.18**
RNA-Seq train LDA	0.652	0.509	0.926	0.928	0.927	3.19
RNA-Seq test FPP	**0.695**	**0.401**	**0.984**	**0.979**	**0.981**	N/A
RNA-Seq test LDA	-0.766	4.121	0.278	0.00 694	0.013 5	N/A

Table 2. Performance.

Dataset	Sample	Domain	Range	Size	Epoch	Batch	Timing(s)
Circle (5D)	3000	5	1	0.14 (MB)	50	50	2.98
Circle (30D)	3000	30	1	0.74 (MB)	50	50	3.39
ICF simulator	1 M	5	15	152.6 (MB)	1	200	24.7
RNA-Seq	801	20 531	1	125.5 (MB)	10	30	1.18
MNIST	60 K	784	1	179.4 (MB)	20	100	17.75
ResNet ImageNet Sub	611 K	2048	1	9.3 (GB)	10	200	184.5
ResNet ImageNet Full	1.28 M	2048	1	19.6 (GB)	10	200	404.8
AlexNet ImageNet Sub	611 K	4096	1	18.7 (GB)	10	200	283.7
AlexNet ImageNet Full	1.28 M	4096	1	39.1 (GB)	10	200	640.1

To quantitatively compare the 2D embedding results between FPP and LDA, we compute 'unsupervised' quality scores (*i.e.* only require one set of labels, instead of both prediction and ground truth) such as *mean silhouette coefficient* (Rousseeuw 1987) (a higher value indicates better cluster separation) and *Davies-Bouldin score* (Davies and Bouldin 1979) (a lower value indicates a more coherent cluster pattern). Alternatively, we can measure the quality of the grouping pattern in 2D by estimate which configuration can be more easily learned by a 2D classifier. To achieve this, we train a kernel support vector machine (Platt *et al* 1999) to identify localized clusters in both of the 2D embeddings obtained from FPP and LDA. We then estimate their quality by examining the predicted and ground truth label via metrics such as *homogeneity*, *completeness* and the *normalized mutual information scores* (Rosenberg and Hirschberg 2007). The results for both types of quantitative analysis are summarized in table 1, in which the more desirable quantities are highlighted in bold font. As we can see all the measures indicate FPP produces 2D embeddings with better cluster separation. For these small datasets, the computation times are comparable for FPP and LDA. However, as illustrated in the following example, FPP can easily handle extremely large dataset that often not viable for traditional methods such as LDA.

The 2D loss function combined with the SGD based implementation allows FPP to scale to data sizes significantly beyond the ability of most traditional projection/dimensionality reduction approaches. In the following example, we use FPP to probe into feature representations of two popular deep learning architecture (AlexNet, ResNet) on the entire ImageNet challenge (Deng *et al* 2009) training dataset, which consists of more than 1.28 million images and 1000 classes. We use the last layer before the *softmax* as the feature representation for both networks resulting in a 2048- and 4096-dimensional feature space for ResNet and AlexNet, respectively. Combined with a large number of samples this results in datasets of more than 40 GB. Despite their massive size, we can generate projections for each of these 1.28 M sample datasets within minutes (between 3 to 10 minutes). The detailed timing results and parameter setups for all our examples are in listed in table 2. For the first five datasets, we compute the results on a laptop with an Intel Core i7-6820HQ processor (2.9 GHz), whereas the last four are computed on a server (due to limited memory on the laptop) with an Intel Xeon E5-2695 processor (2.1 GHz).

For the experiments, we first group the entire 1.28 M images (1000 classes) into 5 coarse categories, namely, 'living thing', 'natural object', 'food', 'artifact', 'misc', where the first 4 categories consists 984 of the 1000 classes available in the imageNet challenge. We then project all the images feature representations with the categories as the label. As shown in figure 6(a), we can see that the AlexNet's representation has trouble distinguishing the purple samples ('misc' category) from the rest, whereas the ResNet's representation, despite being lower-dimensional, can. For visualizing more detailed category separability, we generate 10 categories with more concrete and meaningful labels (*i.e.* 'fish', 'bird', 'mammal', 'invertebrate', 'food', 'fruit', 'vehicle', 'appliance', 'tool', 'instrument'), which consists of 611k images of the 1.28 M. As we can see in figure 6(b), the feature representation of ResNet again seems to be able to better separate these categories

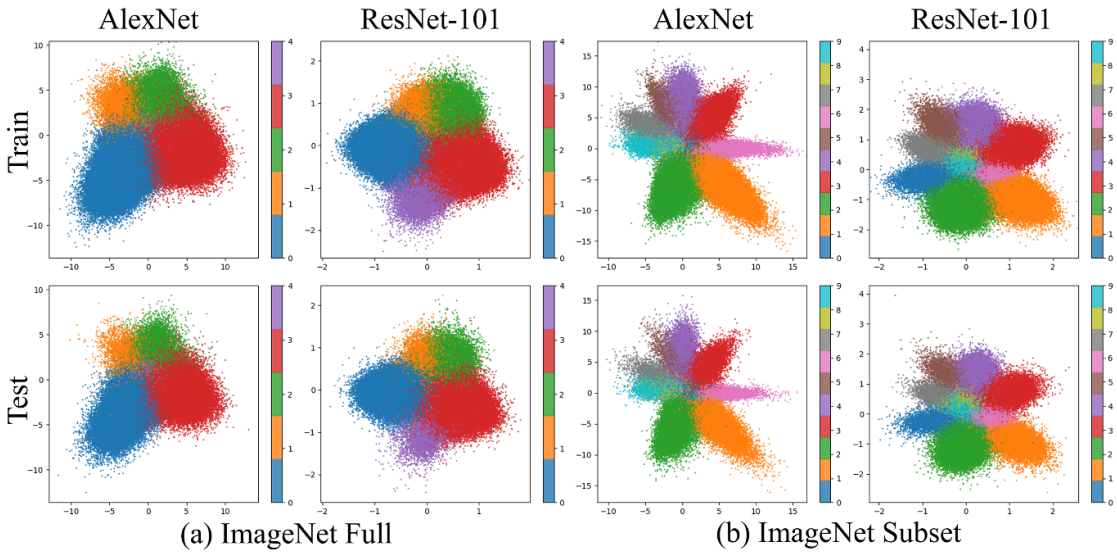


Figure 6. Projections of imageNet image feature representations from different neural network architectures. From the comparison, we can see that the representation produced by ResNet-101 is easier to separate by categories in the linear projection than that of the AlexNet.

compared to AlexNet's representations, which may explain the gap in their predictive performances (AlexNet and ResNet have Top-5 errors of 20.91% and 6.44%, respectively).

5. Conclusion

In this work, we introduce a novel class of linear projection methods for visualizing interesting and interpretable visual patterns of the function in 2D subspaces of the function domain. The combination of linear projection and non-linear pattern searching schemes (*i.e.* a polynomial regressor, or a non-linear classifier in 2D) allows us to exploit our innate ability to perceive complex (and potentially non-linear) visual pattern in 2D while compensating for our inability to comprehend non-linear transformation by focusing only on a linear transformation from high-dimensional space to 2D. The efficient formulation also allows us to easily scale the problem beyond million of samples and tens of thousands of dimensions that is infeasible for most existing dimensionality reduction methods.

Acknowledgement

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Released under LLNL-JRNL-790959.

Data availability

The multivariate physics data is available on https://github.com/rushilanirudh/macc
The MNIST handwritten digit dataset is available on http://yann.lecun.com/exdb/mnist/
The RNA sequence dataset is available on
https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq
The imageNet dataset is available on http://www.image-net.org/

Code availability

The code is available on

ORCID iD

Shusen Liu https://orcid.org/0000-0002-6455-8391

References

Abadi Min*et al* 2016 Tensorflow: Large-scale machine learning on heterogeneous distributed systems 12th USENIX Symposium on Operating Systems Design and Implementation pp 265–283

Carr D B, Littlefield R J, Nicholson W L and Littlefield J S 1987 Scatterplot matrix techniques for large n *J. Am. Stat. Assoc.* **82** 424–36

Chatzimparmpas A, Martins R M and Kerren A 2020 t-visne: Interactive assessment and interpretation of t-SNE projections *IEEE Trans. on Visualization and Computer Graphics* pp 2696–714

Chin W W *et al* 1998 The partial least squares approach to structural equation modeling *Modern Methods Business Res.* **295** 295–336

Davies D L and Bouldin D W 1979 A cluster separation measure *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp 224–7

Deng J, Dong W, Socher R, Li Li-J, Li K and Fei-Fei Li 2009 Imagenet: A large-scale hierarchical image database *2009 IEEE Conference on Computer Vision and Pattern Recognition* IEEE pp 248–55

Donoho D L and Johnstone I M 1989 Projection-based approximation and a duality with kernel methods *The Annals of Statistics* pp 58–106

Espadoto M, Martins R M, Kerren A, Hirata N S T and Telea A C 2019 Towards a quantitative survey of dimension reduction techniques *IEEE Trans. Vis. Comput. Graphics* (https://ieeexplore.ieee.org/abstract/document/8851280)

Fisher R A 1936 The use of multiple measurements in taxonomic problems *Annals Eugenics* **7** 179–88

Friedman J H and Stuetzle W 1981 Projection pursuit regression *J. Am. Stat. Assoc.* **76** 817–23

Gaffney J A *et al* 2020 The jag inertial confinement fusion simulation dataset for multi-modal scientific deep learning *Lawrence Livermore National Laboratory (LLNL) Open Data Initiative* (UC San Diego Library Digital Collections) p 3

Gaffney J, Springer P and Collins G Thermodynamic modeling of uncertainties in NIF ICF implosions due to underlying microphysics models *APS Meeting Abstracts* p 2014 2014

Golub G H and Reinsch C 1971 Singular value decomposition and least squares solutions *Linear Algebra* (Berlin: Springer) pp 134–51

Hardoon D R, Szedmak S and Shawe-Taylor J 2004 Canonical correlation analysis: An overview with application to learning methods *Neural Comput.* **16** 2639–64

Inselberg A and Dimsdale B 1990 Parallel coordinates: a tool for visualizing multi-dimensional geometry *Proc. of the 1st Conference on Visualization'90* pp 361–78 IEEE Computer Society Press

Jolliffe I 2011 *Principal Component Analysis* (Berlin: Springer)

Kruskal J B 1964 Nonmetric multidimensional scaling: a numerical method *Psychometrika* **29** 115–29

Li K-C 1991 Sliced inverse regression for dimension reduction *J. Am. Stat. Assoc.* **86** 316–27

Liu S, Wang B, Bremer P-T and Pascucci V 2014 Distortion-guided structure-driven interactive exploration of high-dimensional data *Computer Graphics Forum* vol 33 pp 101–110

Nonato L G and Aupetit M 2018 Multidimensional projection for visual analytics: Linking techniques with distortions, tasks and layout enrichment *IEEE Trans. Vis. Comput. Graphics* **25** 2650–73

Platt J *et al* 1999 Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods *Advances in Large Margin Classifiers* **10** 61–74 MIT Press Cambridge, MA

Rosenberg A and Hirschberg J 2007 V-measure: A conditional entropy-based external cluster evaluation measure In *Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* pp 410–20

Rousseeuw P J 1987 Silhouettes: a graphical aid to the interpretation and validation of cluster analysis *J. Comput. Appl. Math.* **20** 53–65

Springer P T *et al* 2013 Integrated thermodynamic model for ignition target performance *EPJ Web of Conferences* vol 59 p 04001 EDP Sciences

Tukey J W 1977 *Exploratory Data Analysis* (Reading, MA: Addison-Wesley) vol 2

Xiaofei H and Niyogi P 2004 Locality preserving projections In *Advances in Neural Information Processing Systems* pp 153–60

van der Maaten L and Hinton G 2008 Visualizing data using t-SNE *J. Machine Learning Res.* **9** 2579–605