Semi-Supervised Classification and Visualization of Multi-View Data

Theodoulos Rodosthenous* Vahid Shahrezaei* Marina Evangelou*

Abstract

An increasing number of multi-view data are being published by studies in several fields. This type of data corresponds to multiple data-views, each representing a different aspect of the same set of samples. We have recently proposed multi-SNE, an extension of t-SNE, that produces a single visualisation of multi-view data. The multi-SNE approach provides low-dimensional embeddings of the samples, produced by being updated iteratively through the different data-views. Here, we further extend multi-SNE to a semi-supervised approach, that classifies unlabelled samples by regarding the labelling information as an extra data-view. We look deeper into the performance, limitations and strengths of multi-SNE and its extension, S-multi-SNE, by applying the two methods on various multi-view datasets with different challenges. We show that by including the labelling information, the projection of the samples improves drastically and it is accompanied by a strong classification performance.

Key Words: Data visualisation, Dimensionality reduction, Semi-supervised classification, t-SNE, Multi-view data, Manifold learning

1. Introduction

Multi-view data are usually described as a collection of data taken from different sources on the same samples. It is now very common for multi-view data to be generated in different fields; for example multi-omics datasets in biomedical studies [19], crystal structure data in the field of chemistry [4], data sources in social science [12] and cyber-security [13]. In biomedical studies multiple omics datasets, *e.g. proteomics, genomics, transcriptomics*, are generated on the same individuals. Through these studies the researchers are interested in understanding the relationships between the omics datasets, the underlying biology, and also enhance their relationship with the studied disease (*e.g.* classifying patients as healthy or not). In this manuscript, we focus on the latter, and specifically on the task of classifying samples by utilising the multi-view data. We propose a semi-supervised learning approach, named *S-multi-SNE*, that incorporates the labelling information of the training set alongside the multi-view data (training and test) to visualise all samples and classify the labels of the test set.

S-multi-SNE is an extension of our recent work on multi-view visualisation approach, *multi-SNE* that produces a single representation of the samples by incorporating the information of all data views. *Multi-SNE* is a multi-view extension of the widely used dimensionality reduction approach, t-distributed Stochastic Neighbour Embedding (t-SNE) [23], that has gained great popularity over the last years as it provides a comprehensible low-dimensional projection of the samples in a single-view setting. Multi-SNE was found to have superior performance in the visualisation of the samples and identification of any underlying structure when compared with the competitive extension of t-SNE, named *m-SNE* [25], and other multi-view manifold learning approaches [18].

The proposed adaptation of the multi-SNE approach focuses more in producing a good data visualisation by incorporating the labelling information of the samples, which is constructed as a binary matrix of size $N \times C$, where C presents the total number of classes of

*Department of Mathematics, Imperial College London, South Kensington Campus, SW7 2AZ, UK

the samples and N is the number of samples. The cell (i, j) of the matrix takes the value 1 if sample i belongs to class j and zero otherwise. For example, in a multi-omics experiment on cancer patients and controls, the labelling matrix will have two columns, one column for the cancer patients, and another one for the controls. Similarly, in a cancer subtypes study where the samples are patients with different cancer types (for example [24]), each column of the labelling matrix will correspond to each cancer type.

The proposed approach, *S-multi-SNE*, combines labelling information of the training samples, with the training and test sets of the multi-view data, for classifying the labels of the test samples. This is done by applying the multi-SNE algorithm on the available data and applying a classification algorithm on the projected low-dimensional embeddings produced by the algorithm. *S-multi-SNE* is a transductive algorithm, as it does not build a generic predictive model. Such algorithms tend to make predictions on a specific test set [21]. If a new data point is added to the test set, then the algorithm has to re-run from the beginning to train the model and then to predict the labels. Transductive learning algorithms are preferred when multiple test (query) sets are available with different characteristics.

The *S-multi-SNE* projections are treated as features in the classification algorithm that predicts the classes of the test samples. Different classification algorithms can be utilised for this purpose. Through a series of experiments we illustrate that the K-Nearest Neighbours (KNN) [10] classification algorithm has a good performance. An advantage of KNN is that a good classification score ensures a good visualisation of the data. That is, because KNN separates different classes into neighbourhoods and classifies the samples in the test set by looking at their neighbours [10].

Related work

In the last few years, a number of multi-view semi-supervised learning approaches have been proposed [2, 15, 16, 26]. Bo et al. (2019) [2] conducted a simulation study where they compared the performance of the recently published semi-supervised multi-view classification approaches: Auto-weighted Multiple Graph Learning (AMGL) [16], Multi-view Learning with Adaptive Neighbors (MLAN) [15] and Latent Multi-view Semi-Supervised Classification (LMSSC) [2]. In their study, Bo et al.(2019) demonstrated that LMSSC was superior to the other algorithms, under different scenarios. The LMSSC approach classifies the test samples in two steps: (a) constructs a graph, with samples as nodes and weighted edges based on similarities among all data-views, and (b) uses label propagation to infer the labels on unlabelled samples. Following the results of Bo et al. (2019), we have compared our proposed *S-multi-SNE* approach with the LMSSC approach.

The popularity of t-SNE attracted many researchers who proposed several variations and extensions of the algorithm. Recently, Cheng et al. (2020) proposed St-SNE [5], a supervised extension of t-SNE. Similarly to our proposal, St-SNE considers the labelling information as an additional data-view. In contrast to S-multi-SNE, the unlabelled samples are classified differently. They proposed three strategies for classification, one of which uses Neural Networks (nSt-SNE) and it is similar to the parametric t-SNE [22]. Another strategy (dSt-SNE) implements St-SNE twice to predict the classes of unlabelled samples. The latter strategy was used to compare a single-view approach (St-SNE) against a multi-view approach (S-multi-SNE) and to highlight the benefits of incorporating multiple data-views in the analysis.

A comparative study between S-multi-SNE, LMSSC and St-SNE was conducted to explore and assess their classification performance. All three algorithms were implemented on 10%, 20%, 50%, and 80% of the samples in training, covering both semi-supervised and supervised scenarios. The focus of the comparison with St-SNE lies on 80% training rate, while the emphasis against LMSSC falls on the low training rates (10%, 20%, 50%). We show that S-multi-SNE performs closely to LMSSC and outperforms St-SNE. In particular, S-multi-SNE was superior on 80% training/test rate and it was found to outperform LMSSC on datasets with imbalanced labels or small sample size per class.

In the following section, we describe the proposed *S-multi-SNE* approach and the multi-view datasets used. Through a series of experiments we illustrate and discuss:

- the performance of visualising samples when labelling information is included
- the choice of the classifier algorithm and how it affects the performance of S-multi-SNE
- the performance of S-multi-SNE versus LMSSC and St-SNE
- the performance of S-multi-SNE and LMSSC on datasets with imbalanced labels and small sample size per class

2. Materials and Methods

In this section, multi-SNE is described and its extension, S-multi-SNE is introduced. The classification algorithms and datasets used in this study are then reported.

2.1 Multi-SNE

Suppose that **X** is a multi-view dataset that contains M data-views. Let $X^{(m)} \in \mathbb{R}^{N \times p_m}$ denote the m^{th} data-view, with $\mathbf{x}_i^{(m)}$ being the i^{th} data point of $X^{(m)}$, where $m = 1, \cdots, M$. Let $Y \in \mathbb{R}^{N \times d}$ represent the low-dimensional embedding of the original data obtained as the output of multi-SNE; $\mathbf{y}_i$ is the i^{th} data point of Y and $d = 2$ was set throughout the paper.

For data-view m, multi-SNE measures the probability distribution, $P^{(m)}$, of each data point, $\mathbf{x}_i^{(m)}$, as follows: For every sample i, a sample j is taken as its potential neighbour with probability $p_{ij}^{(m)}$, given by:

$$p_{ij}^{(m)} = \frac{\exp\left(-(d_{ij}^{(m)})^2\right)}{\sum_{k \neq i} \exp\left(-(d_{ik}^{(m)})^2\right)}, \tag{1}$$

where $d_{ij}^{(m)} = \frac{||\mathbf{x}_i^{(m)} - \mathbf{x}_j^{(m)}||^2}{2\sigma_i^2}$ represents the dissimilarity between points $\mathbf{x}_i^{(m)}$ and $\mathbf{x}_j^{(m)}$. The obtained probability distribution $P_i^{(m)} = \sum_j p_{ij}^{(m)}$ has a fixed perplexity, which refers to the effective number of local neighbours. Perplexity is defined as $Perp(P_i^{(m)}) = 2^{H(P_i^{(m)})}$, where $H(P_i^{(m)}) = -\sum_j p_{ij}^{(m)} \log_2 p_{ij}^{(m)}$ is the Shannon entropy of $P_i^{(m)}$, typically taking values between 5 and 50. The results presented in this manuscript are taken with optimized perplexity.

A probability distribution in the low-dimensional space follows Student's t-distribution with one degree of freedom [23] and it is computed as follows:

$$q_{ij} = \frac{(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||\mathbf{y}_k - \mathbf{y}_l||^2)^{-1}},$$

which represents the probability of point i selecting point j as its neighbour.

The Kullback-Leibler divergence (KL-divergence) provides a measure of how different a probability distribution, G is from a second probability distribution, H, denoted by $KL(G||H)$ [14]. If $KL(G||H) = 0$, then the probability distributions G and H are identical. The induced embedding output, $\mathbf{y}_i$, represented by probability distribution, Q, is obtained by minimising the sum of all KL-divergence measures between Q and the distributions of every data-view $m = 1, \cdots, M$. In other words, multi-SNE minimises the cost function given by equation (2).

$$\begin{aligned} C_{multi-SNE} &= \sum_m \sum_i w^{(m)} KL(P_i^{(m)} || Q_i) \\ &= \sum_m \sum_i \sum_j w^{(m)} p_{ij}^{(m)} \log \frac{p_{ij}^{(m)}}{q_{ij}}, \end{aligned} \tag{2}$$

where $w^{(m)}$ provides a weight value for each data-view, and $\sum_m w^{(m)} = 1$. Here, we have taken equal weights on all data-views, *i.e.* $w^{(m)} = \frac{1}{M}, \quad \forall m = 1, \cdots M$.

2.2 S-multi-SNE

The iterative property of multi-SNE provides the option to modify the algorithm in a way that includes the labelling information and consequently make predictions on unlabelled samples. In this section, we propose such a modification, named *S-multi-SNE*. The algorithm of this approach can be found in the supplementary material.

Suppose that M data-views are available, and assume w.l.o.g. that the last data-view, denoted by $X^{(l)} = X^{(M)}$, contains the labelling information in a binary matrix format. For $m = 1, \cdots, M$, let $X_{TR}^{(m)} \subset X^{(m)}$ be the training set (contains information on labelled samples) and $X_{TE}^{(m)} \subset X^{(m)}$ be the test set (with unlabelled samples). The low-dimensional embeddings of the data points in the training set are computed by using all available data-views, including $X_{TR}^{(l)}$. On the other hand, the embeddings of the data points in the test set do not consider $X_{TE}^{(l)}$, since that information is missing ($X_{TE}^{(l)} = \emptyset$).

Let $\mathfrak{I}^{(l)} \in \mathbb{R}^{N \times N}$ be defined by:

$$\mathfrak{I}_{ij}^{(l)} = \begin{cases} 1 & \text{if } \{D_i^{(l)} = 1\} \wedge \{D_j^{(l)} = 1\} \\ 0 & \text{otherwise} \end{cases}$$

where $D^{(l)} \in \mathbb{R}^N$ denotes the *missing data*, defined by:

$$D_i^{(l)} = \begin{cases} 0 & \text{if } \mathbf{x}_i^{(l)} \text{ is missing} \\ 1 & \text{if } \mathbf{x}_i^{(l)} \text{ is observed} \end{cases}$$

The cost function of S-multi-SNE is given by:

$$\begin{aligned} C_{S-multi-SNE} = \Bigg[&\sum_m^{M-1} \sum_i \sum_j w^{(m)} p_{ij}^{(m)} \log \frac{p_{ij}^{(m)}}{q_{ij}} + \\ &+ \mathfrak{I}_{ij}^{(l)} w^{(l)} p_{ij}^{(l)} \log \frac{p_{ij}^{(l)}}{q_{ij}} \Bigg] \end{aligned} \tag{3}$$

In every experiment of this study, the data were normalised via PCA, before the implementation of S-multi-SNE. In particular, for each data-view, the first c principal components (PC) that describe 80% of the variance were taken as input in the algorithm. This dimensionality reduction pre-processing step speeds up the algorithm and suppresses some noise, without distorting the distances between data points. Normalisation via PCA is commonly used as a pre-processing step for t-SNE, since it was implemented in the original proposal of the algorithm [23].

T-SNE and by extension its variations, including S-multi-SNE, require a perplexity value, which needs to be tuned. All projections presented in this manuscript are taken with optimised perplexity, which was selected qualitatively by reviewing the data visualisations of each method on a range of perplexity values, $S = \{2, 10, 20, 50, 80, 100, 200\}$. A quantitative evaluation on the classification task assisted in identifying the optimised perplexity.

2.3 Classification algorithms

The output of S-multi-SNE, $Y \in \mathbb{R}^{N \times d}$, are low-dimensional embeddings of all samples, including the ones in the test set. These projections can then be treated as input features into classification algorithms. In a practical manner, any general-purpose classifier can be used to predict the classes of the unlabelled samples. In this paper, several standard classifiers were explored: (a) Support Vector Machine (SVM) [6], (b) Linear Discriminant Analysis (LDA) [9], (c) Decision Trees (DT) [17], (d) Random Forests (RF) [3], (e) Neural Network, via Multi-Layer Perceptron (NN) [20], and (f) K-Nearest Neighbours (KNN) [10].

Each of these classifiers require tuning of one or more parameters. For example, different kernel functions were explored in SVM, and and different solver functions were assessed in LDA. The number of trees, forests, layers and neighbours were optimised in DT, RF, NN and KNN, respectively. A grid search cross-validation framework was implemented to tune the parameter values in each classifier.

The following steps were taken to test the performance of the classifiers on S-multi-SNE, applied on a multi-view dataset **X**.

1. Randomly split the samples in training/test sets with 10%, 20%, 50% and 80% of samples lying in the training set. The split is performed in proportion to each class size within a dataset.

2. Implement S-multi-SNE on **X**.

3. Implement a classifier algorithm (*e.g.* KNN) on the low-dimensional embeddings produced by S-multi-SNE to classify the samples in the test set.

4. Repeat steps $1 - 3$, for a $N_{iter} = 100$ times.

2.4 Data Description

The aim of the study is to explore the performance of S-multi-SNE and compare it against existing approaches. In order to test the robustness of the algorithm, it is important to explore datasets with distinct attributes. Four real and one synthetic datasets were analysed. The different characteristics (*e.g.* high-dimensionality, heterogeneity, number of data-views, samples and classes) of each dataset allow us to evaluate the methods in a range of real-life situations with noisy data. The real datasets are classified as heterogeneous due to the nature of their data; the synthetic dataset is classified as non-heterogeneous, since it was generated under the same conditions and distributions. The datasets analysed in this paper are described below:

Handwritten Digits [1] **[7]:** Extracted from a collection of Dutch utility maps.
Number of classes: **10** [Handwritten numerals $(0-9)$].
Number of data-views: **6**: (a) Fourier coefficients of the character shapes ($p_1 = \mathit{76}$), (b) profile correlations ($p_2 = \mathit{216}$), (c) Karhunen-Love coefficients ($p_3 = \mathit{64}$), (d) pixel averages in 2 x 3 windows ($p_4 = \mathit{240}$), (e) Zernike moments ($p_5 = \mathit{47}$) and (f) morphological features ($p_6 = \mathit{6}$).
Number of samples: **2000** [*200* patterns per class].

Caltech7 [2] **[8]:** Subset of Caltech-101.
Number of classes: **7** [Pictures of 7 different objects].
Number of data-views: **6**: (a) Gabor ($p_1 = \mathit{48}$), (b) wavelet moments ($p_2 = \mathit{40}$), (c) CENTRIST ($p_3 = \mathit{254}$), (d) histogram of oriented gradients ($p_4 = \mathit{1984}$), (e) GIST ($p_5 = \mathit{512}$), and (f) local binary patterns ($p_6 = \mathit{928}$).
Number of samples: **1474**: Imbalanced dataset with samples per class: {A: 435, B: 798, C: 52, D: 34, E: 35, F: 64, G: 56}.

Cancer Types [3] **[24]:** Multi-omics.
Number of classes: **3** [Cancer types (breast, kidney, lung)].
Number of data-views: **3**: (a) genomics ($p_3 = \mathit{10299}$), (b) epigenomics ($p_2 = \mathit{22503}$) and (c) transcriptomics ($p_3 = \mathit{302}$).
Number of samples: **253**: $\mathit{65}$ patients with breast cancer, $\mathit{82}$ with kidney cancer and $\mathit{106}$ with lung cancer.

Reuters [4] **[1]:** Text documents.
Number of classes: **6** [E21, CCAT, M11, GCAT, C15, ECAT].
Number of data-views: **5**: Words from the original documents (English) and from four translations ((a) Italian, (b) French, (c) German and (d) Spanish). All five data-views contain 2000 features (words).
Number of samples: **1200** : [*200* documents per class]. .

Noisy Data-view Synthetic data (NDS) [18]: Synthetic dataset.
Number of classes: **3** [$\mathbf{A}, \mathbf{B}, \mathbf{C}$].
Number of data-views: **4**: $p_1 = 100, p_2 = 100, p_3 = 100, p_4 = 1000$
Number of samples: **300** : [*100* samples per class].

The synthetic dataset (NDS) aims to justify the use of the algorithm and its ability to capture the true underlying classes of the samples, even when they are not well-represented in each data-view. The data follow the noisy data-view scenario described by Rodosthenous et al. (2021) [18] and were generated as follows. Each sample follows a normal distribution with mean μ and standard deviation $\sigma = 1$. To distinguish the classes, different μ values were used for each data-view. Further, noise ($\epsilon \sim \mathcal{N}(\mu_\epsilon, \sigma_\epsilon)$) was added to increase randomness within the data-views. Lastly, polynomial functions were applied on the samples to express non-linearity and ensure that linear dimensionality reduction methods (*e.g.* PCA) would not succeed in identifying the classes.

In NDS, the first data-view separates only cluster **A** from the others, the second view separates only cluster **B** and the third view separates only cluster **C**. The first three data-views have $p_v = 100$ features. The last data-view represents a noisy data-view (all data points lie in one cluster) with $p_v = 1000$ features to intensify the noise. The data structure in NDS highlights the importance of multi-view analysis, since each data-view describes a distinct clustering, none of which describes accurately the synthetic truth. An effective multi-view algorithm would distinguish the three clusters while it avoids the noise of the

[1] https://archive.ics.uci.edu/ml/datasets/Multiple+Features

[2] https://github.com/yeqinglee/mvdata

[3] http://compbio.cs.toronto.edu/SNF/SNF/Software.html

[4] https://github.com/lzu-cvpr/multiview-learning/blob/master/multiview_DataSets.md

4^{th} data-view.

2.5 Performance evaluation

The classification performance of each method, was evaluated by three common measures: (A) Accuracy, (B) Precision, and (C) Recall. Let $T \in \mathbb{R}^{N \times k}$ denote the true classes of a dataset with N samples. For each $1 \leq i \leq N$, $T_i \in \mathcal{T} = \{0, 1\}^k$, with 1 referring to its true class. Similarly, let $P \in \mathbb{R}^{N \times k}$ denote the predicted classes.

Accuracy: The proportion of correctly predicted labels to the total number of labels (predicted and actual).

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \frac{|T_i \cap P_i|}{|T_i \cup P_i|} \tag{4}$$

Precision: The proportion of correctly predicted labels to the total number of actual labels.

$$\text{Precision} = \frac{1}{N} \sum_{i=1}^{N} \frac{|T_i \cap P_i|}{|T_i|} \tag{5}$$

Recall: The proportion of correctly predicted labels to the total number of predicted labels.

$$\text{Recall} = \frac{1}{N} \sum_{i=1}^{N} \frac{|T_i \cap P_i|}{|P_i|} \tag{6}$$

All three evaluation measures lie in the range $[0, 1]$, with 0 referring to a complete misclassification of the unlabelled samples, while 1 refers to a classification that is perfectly aligned with the ground truth.

3. Results

In this section, the performance results of S-multi-SNE are presented. Section 3.1 compares several classification algorithms to assess whether the choice of a classifier influences the performance of the algorithm. In Section 3.2, S-multi-SNE is compared against LMSSC and St-SNE to explore the performance of our proposal against recent related semi-supervised algorithms. Both LMSSC and St-SNE require parameter tuning. The results presented in this section are obtained with optimal tuning parameters, which were selected according to their respective publications. Section 3.3 investigates two specific scenarios, commonly observed in real scenarios: (i) imbalanced data, and (ii) datasets with small sample size.

3.1 Classifier selection

The different classification algorithms mentioned in Section 2.3 are assessed on their predictive performances to evaluate the impact of the classifier on S-multi-SNE.

Following the process described in Section 2.3, all six classifiers performed well on every dataset, with SVM, RF and KNN being the most consistent (Table 1 depicts their evaluation performance with 50% training samples). In particular, KNN performed equally well, or outperformed the other classifiers on all datasets.

	Classifier	Handwritten digits	Caltech7	Cancer types	Reuters	NDS
Accuracy	SVM	0.97 (0.005)	0.94 (0.009)	0.90 (0.024)	0.74 (0.029)	0.95 (0.005)
	LDA	0.97 (0.008)	0.90 (0.018)	0.89 (0.027)	0.54 (0.093)	0.92 (0.005)
	DT	0.95 (0.011)	0.93 (0.011)	0.87 (0.029)	0.72 (0.025)	0.91 (0.013)
	RF	0.97 (0.006)	**0.95 (0.007)**	**0.91 (0.024)**	0.73 (0.021)	0.92 (0.007)
	NN	0.73 (0.060)	0.92 (0.009)	0.89 (0.025)	0.71 (0.041)	0.92 (0.006)
	KNN	**0.98 (0.005)**	**0.95 (0.009)**	0.90 (0.023)	**0.75 (0.021)**	**0.96 (0.004)**
Precision	SVM	0.98 (0.004)	0.96 (0.010)	**0.96 (0.019)**	**0.86 (0.031)**	**0.98 (0.004)**
	LDA	0.98 (0.005)	0.94 (0.020)	0.95 (0.021)	0.70 (0.095)	0.94 (0.004)
	DT	0.97 (0.007)	0.96 (0.011)	0.93 (0.028)	0.84 (0.043)	0.94 (0.009)
	RF	0.98 (0.005)	0.97 (0.007)	0.95 (0.020)	0.85 (0.020)	0.96 (0.005)
	NN	0.83 (0.061)	0.94 (0.014)	0.95 (0.020)	0.84 (0.046)	0.94 (0.005)
	KNN	**0.99 (0.004)**	**0.97 (0.008)**	**0.96 (0.017)**	**0.86 (0.024)**	**0.98 (0.004)**
Recall	SVM	**0.98 (0.005)**	**0.98 (0.007)**	**0.94 (0.021)**	0.84 (0.030)	**0.98 (0.003)**
	LDA	**0.98 (0.007)**	0.96 (0.011)	0.93 (0.025)	0.71 (0.098)	0.97 (0.003)
	DT	0.97 (0.010)	0.97 (0.011)	0.93 (0.028)	0.83 (0.035)	0.97 (0.010)
	RF	**0.98 (0.005)**	**0.98 (0.011)**	0.93 (0.021)	**0.85 (0.021)**	0.97 (0.005)
	NN	0.87 (0.092)	**0.98 (0.011)**	0.93 (0.023)	0.82 (0.043)	0.97 (0.004)
	KNN	**0.98 (0.004)**	**0.98 (0.011)**	**0.94 (0.019)**	**0.85 (0.022)**	**0.98 (0.002)**

Table 1: **Classifiers performance.** The mean (and standard deviation) accuracy, precision and recall of SVM, LDA, DT, RF, NN and KNN with 50% training bootstrap resamples from the handwritten digits, caltech7, cancer types, Reuters and NDS datasets. For each evaluation measure, the classifier with the best performance on each dataset is highlighted with **bold**.

On the other hand, LDA had the most inconsistent performance across the datasets. This observation is particularly noted on Reuters and caltech7, for which classification is a more challenging task than for the other datasets. Further, DT and NN had the highest variability in performance on all three measures.

Additionally, the performance of the algorithms improve as the number of training samples increases, and all classifiers tend to have similar performances (Figure 1). This observation does not come as a surprise, since it is reasonable to expect better classification performance with a larger training set.

Overall, we found KNN to have the most consistently good performance. For that reason, the classification task for the experiments that follow was performed by KNN. Based on the foundation of KNN, *i.e.* finding neighbourhoods in the sample space, and since the embeddings are two-dimensional, there is a direct relationship between its quantitative (classification) and qualitative (visualisation) performances. A good quantitative performance from KNN suggests a good two-dimensional projection of the data, since nearest neighbours belong to the same class.

3.2 Semi-supervised classification

A comparative study between S-multi-SNE, LMSSC and St-SNE was conducted to assess the performance of S-multi-SNE against similar state-of-the-art algorithms. The aim of this section is two-fold. The comparison with LMSSC demonstrates the performance of S-multi-SNE against a state-of-the-art multi-view semi-supervised classification method, while the comparison with St-SNE highlights the benefits of incorporating multiple data-views in contrast to single-view classification and visualisation.

S-multi-SNE outperformed St-SNE in any combination of datasets and training rates

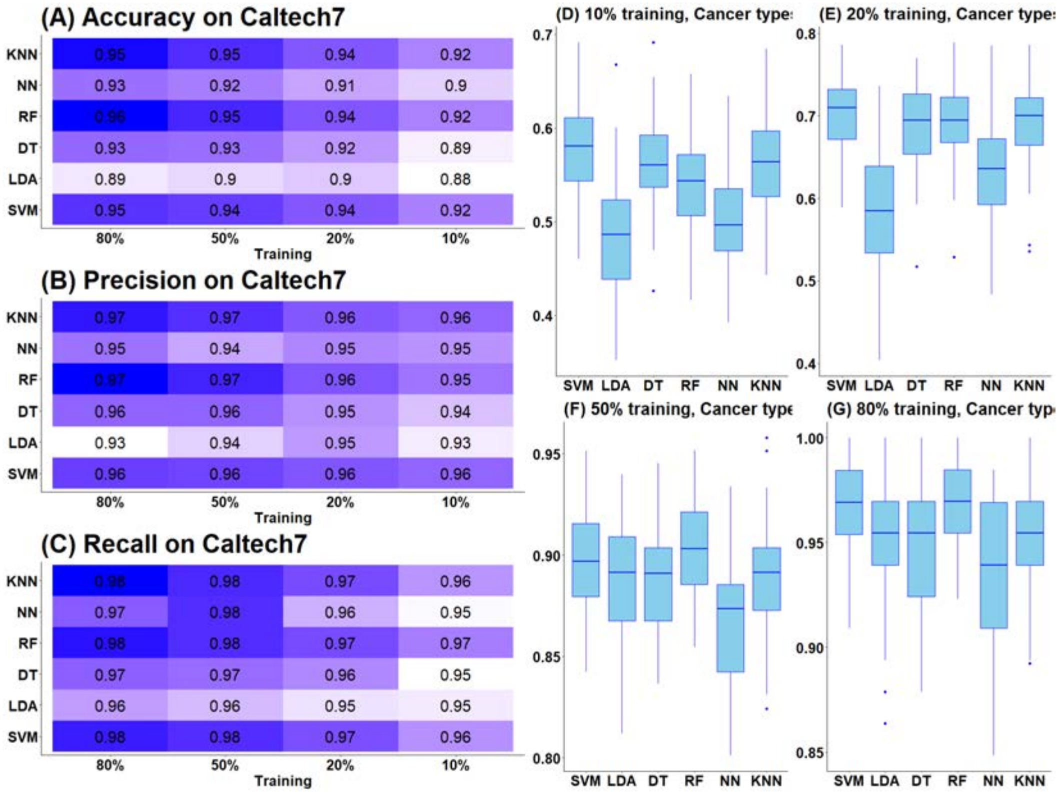

Figure 1: **Classifiers performance on caltech7 and cancer types.** (A-C) Heatmaps of accuracy, precision and recall, respectively, on caltech7 with different training samples. (D-G) Box-plots of accuracy on cancer types with 10%, 20%, 50%, 80% training rates, respectively. On each plot, the performances of all six classifiers are depicted.

(Table 2). St-SNE is not a semi-supervised approach and a good performance on low training rates was not expected. On 80% training rate, the performance of St-SNE did not match the ones from LMSSC or S-multi-SNE which can be explained by the lack of multiple data-views. Concatenating the features of all data-views before implementing St-SNE does not improve its performance. Fu et al. (2008) [11] argue that this is because the information conveyed by different features is not equally represented, since the data-views are described by different data distributions and variation patterns.

As expected, the accuracy of all methods increases when more training samples are available (Table 2). On handwritten digits, Reuters and cancer types, and on low training rates $(10\%, 20\%, 50\%)$ the performance of S-multi-SNE was slightly lower, but still comparable to LMSSC. However, with 80% training rate, S-multi-SNE surpasses the performance of LMSSC. The performance of the methods is influenced by the quality of the data, especially on low training rates (Figures 2 and 3). When training is performed on just 10% of the samples, S-multi-SNE projects the unlabelled samples in Reuters mostly as noise instead of signal (Figure 2 in supplementary material), whereas in handwritten digits, the distinction between classes is successfully achieved (Table 2 and Figure 2).

The analysis on cancer types agrees with the conclusions made on handwritten digits and Reuters. On the other hand, S-multi-SNE on caltech7 outperforms LMSSC on all training rates (Table 2). The split between training and test was performed proportionally to the class sizes. This means that with 10% training, we would only have $4 - 6$ labelled samples for five out of seven classes. S-multi-SNE overcomes this challenge with a better

Dataset	Algorithm	Training rate 10%	Training rate 20%	Training rate 50%	Training rate 80%
Handwritten	S-multi-SNE*	0.952 (0.016)	0.966 (0.008)	0.983 (0.004)	**0.991 (0.004)**
	LMSSC	**0.978 (0.002)**	**0.983 (0.003)**	**0.989 (0.002)**	**0.991 (0.004)**
	BSV St-SNE*	0.682 (0.022)	0.745 (0.027)	0.803 (0.025)	0.866 (0.014)
	Concat. St-SNE*	0.713 (0.026)	0.812 (0.019)	0.855 (0.022)	0.919 (0.020)
Reuters	S-multi-SNE*	0.554 (0.019)	0.632 (0.019)	0.767 (0.017)	**0.906 (0.018)**
	LMSSC	**0.589 (0.025)**	**0.654 (0.022)**	**0.857 (0.017)**	0.899 (0.012)
	BSV St-SNE*	0.173 (0.058)	0.281 (0.030)	0.498 (0.028)	0.657 (0.051)
	Concat. St-SNE*	0.175 (0.54)	0.298 (0.041)	0.526 (0.032)	0.753 (0.048)
Caltech7	S-multi-SNE*	**0.924 (0.010)**	**0.935 (0.006)**	**0.961 (0.007)**	**0.981 (0.008)**
	LMSSC	0.829 (0.040)	0.852 (0.019)	0.878 (0.011)	0.889 (0.011)
	BSV St-SNE*	0.768 (0.014)	0.790 (0.19)	0.823 (0.022)	0.878 (0.20)
	Concat. St-SNE*	0.798 (0.016)	0.817 (0.14)	0.845 (0.015)	0.890 (0.16)
Caltech7-balanced	S-multi-SNE*	**0.928 (0.012)**	**0.950 (0.007)**	**0.977 (0.006)**	**0.991 (0.006)**
	LMSSC	0.629 (0.025)	0.719 (0.032)	0.804 (0.027)	0.843 (0.047)
	BSV St-SNE*	0.492 (0.005)	0.567 (0.006)	0.722 (0.006)	0.868 (0.006)
	Concat. St-SNE*	0.515 (0.010)	0.681 (0.009)	0.819 (0.011)	0.910 (0.007)
CancerTypes	S-multi-SNE*	0.661 (0.042)	0.769 (0.030)	0.914 (0.024)	**0.977 (0.017)**
	LMSSC	**0.783 (0.036)**	**0.883 (0.038)**	**0.957 (0.015)**	0.973 (0.012)
	BSV St-SNE*	0.318 (0.044)	0.568 (0.021)	0.688 (0.034)	0.792 (0.028)
	Concat. St-SNE*	0.290 (0.052)	0.442 (0.054)	0.656 (0.055)	0.774 (0.048)

Table 2: **Semi-supervised classification.** The mean (and standard deviation) accuracy on bootstrap resamples with different training rates from handwritten digits, Reuters, caltech7 and cancer types data. **Bold** highlights the method with the best performance on each training rate within each dataset. *BSV St-SNE* refers to the best single-view performance by St-SNE, while *Concat. St-SNE* implements St-SNE on the concatenated features of all data-views.

accuracy than LMSSC. This observation could suggest that S-multi-SNE is more robust than LMSSC on imbalanced datasets.

To investigate this claim further, a balanced subset of caltech7 was created, by reducing the sample size of classes *A* and *B* to 50 samples. Even though this under-sampling process defeats the obstacle of imbalanced samples, a new challenge arises; overall, a small sample size per class is observed. This new challenge reduced the accuracy of LMSSC, while the performance of S-multi-SNE was unaffected (agrees with the performance of multi-SNE on a similar experiment [18]). This observation could suggest that S-multi-SNE can be used more effectively than LMSSC when the dataset contains a small sample size per class.

3.3 Imbalanced and small sample size

The comparison study between S-multi-SNE and LMSSC in the previous section and specifically on caltech7 dataset, suggests that the former algorithm is more effective than the latter, when the dataset has imbalanced labels or each class has a small number of samples. To explore these two hypotheses further, two subsets of NDS were taken: (A) NDS_{im}: Represents an imbalanced dataset, with 150 samples; all 100 samples from class A were part of the subset, while 20 and 30 samples were randomly selected from classes B and C, respectively. (B) NDS_{sss}: Represents a small sample size dataset, with 30 samples; 10 samples per class were randomly selected.

S-multi-SNE outperformed LMSSC on the synthetic dataset and its subsets (Table 3). This performance can be explained by the noise of the 4^{th} data-view, which had a bigger

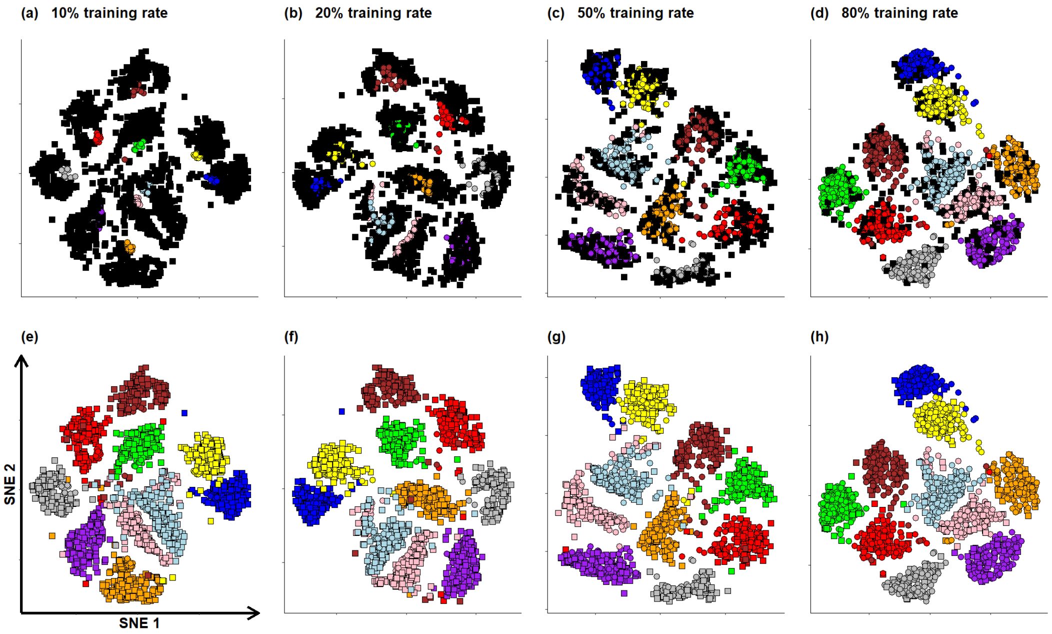

Figure 2: **Data visualisation of handwritten digits with different training rates. (a-d)** Unlabelled samples are presented with black squares. **(e-h)** True labels. The training rates are as follows: **(a)**,**(e)** 10%, **(b)**,**(f)** 20%, **(c)**,**(g)** 50%, and **(d)**,**(h)** 80%.

influence on LMSSC than on S-multi-SNE. Both algorithms had a similar and comparable classification scores on both NDS_{im} and NDS_{sss}. On all scenarios, S-multi-SNE was slightly more accurate than LMSSC, except on NDS_{sss} with 50% training rate.

Throughout all experiments, S-multi-SNE classified the test samples more accurately, when 80% of the samples were in training, but for the remaining training rates, the performances of S-multi-SNE and LMSSC were interchangeable. Although, none of the algorithms showed an evident advantage over the other in the classification task, S-multi-SNE has the benefit of providing an auxiliary comprehensible projection of all samples. This feature may be desirable by researchers who want to explore visually the classes within their data, in addition to the corresponding classification predictions.

Dataset	Algorithm	Training rate 10%	Training rate 20%	Training rate 50%	Training rate 80%
NDS	S-multi-SNE	**0.884 (0.03)**	**0.903 (0.02)**	**0.960 (0.09)**	**0.984 (0.03)**
	LMSSC	0.714 (0.05)	0.733 (0.04)	0.887 (0.07)	0.923 (0.05)
NDS_{im}	S-multi-SNE	**0.686 (0.03)**	**0.738 (0.04)**	**0.775 (0.06)**	**0.833 (0.04)**
	LMSSC	0.669 (0.02)	0.670 (0.03)	0.767 (0.05)	0.800 (0.09)
NDS_{sss}	S-multi-SNE	**0.354 (0.03)**	**0.455 (0.07)**	0.589 (0.08)	**0.797 (0.13)**
	LMSSC	0.345 (0.06)	0.412 (0.08)	**0.673 (0.05)**	0.705 (0.09)

Table 3: **Imbalanced and small sample size classification.** The mean (and standard deviation) accuracy on bootstrap resamples with different training rates on NDS, NDS_{im} and NDS_{sss}. **Bold** highlights the algorithm with the best performance on each training rate within each dataset.

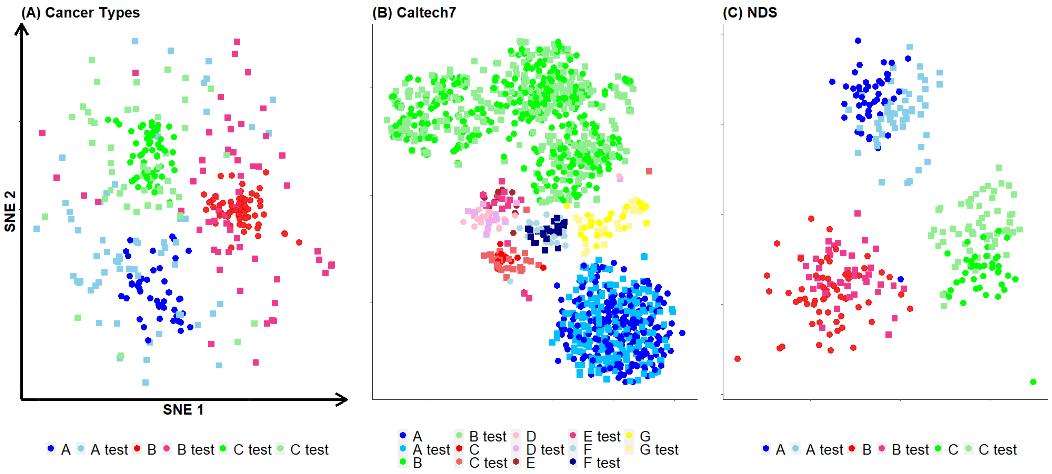

Figure 3: **Data visualisation with 50% training samples.** (A) Cancer types, (B) caltech7 and (C) NDS projections of S-multi-SNE. The training samples are presented with circles and test samples with squares.

4. Discussion

In this work, we propose S-multi-SNE, a semi-supervised learning algorithm for data visualisation and classification. S-multi-SNE produces low-dimensional projections which are used as input features in a classification algorithm to classify the test samples. Although, we found that the classifiers do not have a big effect on the performance of S-multi-SNE, KNN produced the most consistently good predictions out of all six standard classification algorithms tested in this manuscript. Compared against a state-of-the-art multi-view semi-supervised classification approach, S-multi-SNE performed equally well. Specifically, it outperformed LMSSC on caltech7, synthetic dataset NDS, and their subsets, which cover two scenarios: (a) Balanced against imbalanced samples, and (b) small number of samples per class. In addition to its strong classification performance, S-multi-SNE has the desirable feature of producing a comprehensible projection that splits all samples (training and test) to their corresponding classes. On all datasets, S-multi-SNE outperformed a recently proposed supervised variation of t-SNE. This comparison emphasizes on the benefits and importance of performing multi-view analysis over single-view, when available.

Although it was not tested explicitly in this study, S-multi-SNE can be applied on single-view data as well. In this scenario, two data-views would be considered: the first (*i.e.* $X^{(1)}$) would represent the single-view dataset and the second would contain the labelling information (*i.e.* $X^{(2)} = X^{(l)}$).

In some real datasets, it is possible to have samples with missing information in one or more data-views. This can be a result of technical, human or other errors. For example, in a study on genomics, transcriptomics and epigenomics, experimentalists may be unable to get the epigenomics measurements from several patients, but they can have transcriptomics and genomics data. Note that the entire information of a sample would be missing, and not just some values from selected features that got lost. In such situations, a sample with missing information is often entirely excluded from a multi-view analysis, since such analyses require the same number of samples from all data-views. The algorithm of S-multi-SNE can be generalised to allow the analysis of data-views with missing samples. We refer to this generalisation as *G-multi-SNE* and its cost function is given by:

$$C_{G-multiSNE} = \sum_m \sum_i \sum_j \mathfrak{I}_{ij}^{(m)} w^{(m)} p_{ij}^{(m)} \log \frac{p_{ij}^{(m)}}{q_{ij}}, \tag{7}$$

where $\mathfrak{I}_{ij}^{(m)} = \mathbb{1}\{\{D_i^{(m)} = 0\} \wedge \{D_j^{(m)} = 0\}\}$ and $D_i^{(m)} = \mathbb{1}\{\mathbf{x}_i^{(m)} \text{ is missing}\} \in \mathbb{R}^N, \quad \forall m = 1, \cdots M$.

G-multi-SNE, has the potential of increasing the overall sample size, by including, instead of ignoring, samples with missing information. By implementing this approach, researchers could classify the unlabelled samples and at the same time visualise them along with labelled samples and samples with missing information (a visualisation example of G-multi-SNE applied on NDS is displayed in supplementary material).

In this study, we have shown that S-multi-SNE can perform comparably well with a state-of-the-art semi-supervised multi-view classification method, while producing a comprehensive visualisation on two dimensions.

Reproducibility

The public multi-view datasets used in this manuscript can be found by following the links provided in the main body of the paper. We refer the readers to follow the code and functions provided in the link below to reproduce the findings of this paper: `https://github.com/theorod93/S_multi_SNE`. The `R` package `multiSNE` contains the code and functions required to run both multi-SNE and S-multi-SNE. It can be installed through `GitHub` (and `devtools`) from the repository found in: `https://github.com/theorod93/multiSNE`.

References

[1] Massih R. Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views -an application to multilingual text categorization. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, page 28–36, 2009.

[2] Xiaofan Bo, Zhao Kang, Zhitong Zhao, Yuanzhang Su, and Wenyu Chen. Latent multi-view semi-supervised classification. In *Proceedings of The Eleventh Asian Conference on Machine Learning*, volume 101, pages 348–362, 2019.

[3] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] Qing-di Cheng, Hsiang-Yu Chung, Robin Schubert, Shih-Hsuan Chia, Sven Falke, Celestin Nzanzu Mudogo, Franz X. Kartner, Chang Guoqing, and Betzel Christian. Protein-crystal detection with a compact multimodal multiphoton microscope. *Communications Biology*, 3, 2020.

[5] Yichen Cheng, Xinlei Wang, and Yusen Xia. Supervised t-distributed stochastic neighbor embedding for data visualization and classification. *INFORMS Journal on Computing*, 2020.

[6] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20, 1995.

[7] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[8] Li Fei-Fei, Rob Fergus, and Perona Pietro. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[9] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[10] Evelyn Fix and Hodges J. L. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):238–247, 1989.

[11] Yun Fu, Liangliang Cao, Guodong Guo, and Thomas S. Huang. Multiple feature fusion by subspace learning. In *Proceedings of the 2008 International Conference on Content-Based Image and Video Retrieval*, page 127–134, 2008.

[12] Grayson N. Holmbeck, Susan T. Li, Jennifer Verrill Schurman, Deborah Friedman, and Rachael Millstein Coakley. Collecting and Managing Multisource and Multimethod Data in Studies of Pediatric Populations. *Journal of Pediatric Psychology*, 27(1):5–18, 2002.

[13] Alexander D. Kent. Cybersecurity Data Sources for Dynamic Network Research. In *Dynamic Networks in Cybersecurity*, 2015.

[14] Solomon Kullback and Richard A. Leibler. On information and sufficiency. 22:79–86, 1951.

[15] Feiping Nie, Guohao Cai, and Xuelong Li. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, page 2408–2414, 2017.

[16] Feiping Nie, Jing Li, and Xuelong Li. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, page 1881–1887, 2016.

[17] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[18] Theodoulos Rodosthenous, Vahid Shahrezaei, and Marina Evangelou. Multi-view Data Visualisation via Manifold Learning.

[19] Theodoulos Rodosthenous, Vahid Shahrezaei, and Marina Evangelou. Integrating multi-omics data through sparse canonical correlation analysis for the prediction of complex traits: A comparison study. *Bioinformatics*, 36(17):4616–4625, 2020.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, page 318–362, 1986.

[21] Ana Stanescu, Karthik Tangirala, and Doina Caragea. Study of transductive learning and unsupervised feature construction methods for biological sequence classification. ASONAM '16, page 999–1006, 2016.

[22] Laurens van der Maaten. Learning a parametric embedding by preserving local structure. In David van Dyk and Max Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5, pages 384–391, 2009.

[23] Laurens van der Maaten and Geoffrey Hinton. Visualising data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[24] Bo Wang, Aziz M. Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, 11(3):333–337, 2014.

[25] Bo Xie, Yang Mu, Dacheng Tao, and Kaiqi Huang. m-sne: multiview stochastic neighbor embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41:1088–1096, 2011.

[26] Y. Yang, D. C. Zhan, Y. F. Wu, Z. B. Liu, H. Xiong, and Y. Jiang. Semi-supervised multi-modal clustering and classification with incomplete modalities. *IEEE Transactions on Knowledge and Data Engineering*, 33:682–695, 2021.