

# IVFS: Simple and Efficient Feature Selection for High Dimensional Topology Preservation

Xiaoyun Li, Chengxi Wu, Ping Li

Cognitive Computing Lab  
Baidu Research

10900 NE 8th ST. Bellevue WA, 98004, USA  
{lixiaoyun996, wuchenxi2013, pingli98}@gmail.com

## Abstract

Feature selection is an important tool to deal with high dimensional data. In unsupervised case, many popular algorithms aim at maintaining the structure of the original data. In this paper, we propose a simple and effective feature selection algorithm to enhance sample similarity preservation through a new perspective, topology preservation, which is represented by persistent diagrams from the context of computational topology. This method is designed upon a unified feature selection framework called IVFS, which is inspired by random subset method. The scheme is flexible and can handle cases where the problem is analytically intractable. The proposed algorithm is able to well preserve the pairwise distances, as well as topological patterns, of the full data. We demonstrate that our algorithm can provide satisfactory performance under a sharp sub-sampling rate, which supports efficient implementation of our proposed method to large scale datasets. Extensive experiments validate the effectiveness of the proposed feature selection scheme.

## Introduction

High dimensional data becomes more and more common in machine learning applications, e.g., computer vision, natural language processing and gene selection. In many cases, the curse of dimensionality leads to costly computation and a less comprehensible model. Therefore, feature selection is one of the standard data preprocessing methods. The most widely used approach is filter method, where each feature is individually assigned a score according to some statistical measure, and those with highest scores are selected. Supervised filter methods include t-test (Guyon and Elisseeff 2003), mutual information (Peng, Long, and Ding 2005), correlation (Yu and Liu 2003), etc. Meanwhile, there are many unsupervised filter algorithms based on similarity and manifold preservation. LaplacianScore algorithm (He, Cai, and Niyogi 2006) are proposed to choose features according to a nearest neighbor graph. Features that are smoothest on the graph are regarded as most capable to represent the local manifold structure of full data. Spectral Feature Selection (SPEC) (Zhao and Liu 2007) aims at separating the samples

into clusters using the spectrum of pairwise similarity graph. Multi-cluster Feature Selection (MCFS) (Cai, Zhang, and He 2010) considers combining spectral embedding and  $l_1$ -regularized regression by assuming a multi-clustered structure of the data.

Although filter methods are fast to implement, the performance can sometimes be unsatisfactory. In recent years, many learning-based embedded feature selection methods are proposed. The idea of most embedded approaches is to optimize a selection (or weight) matrix with some sparsity regularization. The objective function can be designed to achieve similarity preservation. For instance, Similarity Preserving Feature Selection (SPFS) (Zhao et al. 2013) learns a linear transformation  $W$  by minimizing

$$\min_W \|XW(XW)^T - S\|_F^2 + \lambda \|W\|_{2,1}, \quad (1)$$

where  $X$  is the data,  $S$  is the sample similarity matrix and  $W$  is weight matrix. Features with largest row  $l_2$  norm of  $W$  are selected. SPFS mainly considers preserving similarity globally. An improved version of SPFS is called Global and Local Structure Preservation Feature Selection (GLSPFS) (Liu et al. 2014), whose objective modifies (1) as

$$\min_W \|XW(XW)^T - S\|_F^2 + \mu \cdot \text{tr}(W^T X^T L X W) + \lambda \|W\|_{2,1},$$

where  $L$  is a locality representation graph. This way, local structure is also considered. Other unsupervised embedded algorithms include NDFS (Li et al. 2012), RUFs (Qian and Zhai 2013), FSASL (Du and Shen 2015), SOGFS (Nie, Zhu, and Li 2016), AHLFS (Zhu et al. 2017), UPFS (Li et al. 2018b), DGUFS (Guo and Zhu 2018), etc.

Typically, existing similarity preserving feature selection schemes face two major challenges:

- **High dimensionality.** As will be shown in our experiments, many methods actually perform not as good in high dimensions in terms of distance preserving, possibly due to lack of consideration of feature interactions, or huge number of parameters to optimize. This may result in non-robust or unstable solutions.
- **Large sample size.** In most of the embedded methods introduced above (GLSPFS, SOGFS, etc.), the algorithm involves (repeatedly) matrix decomposition and/or inverse, which runs very slowly with large sample size.

## Topology preserving feature selection

Many embedded methods are associated with clustering or nearest neighbor graph embedding, which in some sense address more on the local manifold structure. In this paper, we look at similarity preservation problem from a new view: topology. In recent years, topological data analysis (TDA) has been shown as a powerful tool for various learning tasks (Hofer et al. 2017). The merit of TDA comes from its capability to encode all the topological information of a point set  $X$  by *persistent diagram*, denoted as  $\mathcal{D}(X)$  herein. In Euclidean space,  $\mathcal{D}(X)$  is tightly related to pairwise sample distances. This motivates us to consider similarity preservation from a TDA perspective. If we hope to select covariates  $X_{\mathcal{F}}$  to preserve the topological information of the original data, the persistent diagram  $\mathcal{D}(X_{\mathcal{F}})$  generated by  $X_{\mathcal{F}}$  should be close to the original diagram  $\mathcal{D}(X)$ . More precisely, TDA offers a new space (of persistent diagrams) in which we compare and preserve the distances. We refer this property as “topology preservation”, which is very important especially when applying TDA after feature selection. In existing literature, however, topology preservation has not been considered yet.

## Our contributions

We develop a simple but powerful filter based feature selection method that achieves topology preservation:

- We propose a unified scheme called Inclusion Value Feature Selection (IVFS) that extends the idea of random subset evaluation to unsupervised case. It generalizes to several well-known supervised methods, i.e., random forest (Breiman 2001) and random KNN (Li, Harner, and Adjeroh 2011; Räsänen and Pohjalainen 2013).
- We use IVFS to achieve topology (persistent diagram) preserving feature selection, based on theories from TDA. By reinforcing the scalability, this algorithm not only perform very well in high dimensions, but also run efficiently with large sample size.
- We conduct extensive experiments on high dimensional datasets to justify the effectiveness of IVFS. The results suggest that IVFS is capable of preserving the exact topological signatures (as well as the pairwise distances), making it a suitable tool for TDA and many other applications.

## IVFS: A Unified Feature Selection Scheme

First, we provide a unified framework based on random subset evaluation, which is flexible, universally applicable, and generalizes to several existing algorithms. In this section, we provide a detailed discussion on this general scheme.

### Problem formulation and notations

The data matrix is denoted by  $X \in \mathbb{R}^{n \times d}$  with  $n$  samples and  $d$  covariates.<sup>1</sup> Our goal is to find the best  $d_0 < d$  features according to some criteria, associated with a loss function  $\mathcal{L}$ . Denote  $\mathcal{I} = \{1, 2, \dots, d\}$  the indices of all features, and

<sup>1</sup>For the ease of presentation, we also use  $X$  to represent data  $(X, Y)$  if the problem is supervised, where  $Y$  is the label vector.

$\mathcal{M}_{\tilde{d}} = \{\sigma \in \mathcal{I}^{\tilde{d}} : \sigma_i \neq \sigma_j \text{ for } \forall i \neq j\}$  the set of all size- $\tilde{d}$  subsets of  $\mathcal{I}$ . The goal is to minimize the objective function

$$\min_{\mathcal{F} \in \mathcal{M}_{d_0}} \mathcal{L}_{\mathcal{F}} \triangleq \min_{\mathcal{F} \in \mathcal{M}_{d_0}} \mathcal{L}(X_{\mathcal{F}}). \quad (2)$$

Our algorithm relies on repeatedly sampling random subset  $\tilde{\mathcal{F}}$  of arbitrarily  $\tilde{d}$  features (not necessarily equal to  $d_0$ ), equipped with a subset score function  $s(\tilde{\mathcal{F}}; X) : \mathbb{R}^{\tilde{d}} \mapsto \mathbb{R}^{\tilde{d}}$  which assigns score to each selected feature by evaluating the chosen random subset. In principle, a high score should correspond to a small loss.

## IVFS: a filter method for feature selection

The individual feature score, which serves as the filter of the unified selection scheme, essentially depends on the subset score function  $s(\cdot)$  defined above.

**Definition 1.** Suppose  $1 \leq \tilde{d} \leq d$ . The **Inclusion Value** of feature  $f \in \mathcal{I}$  at dimension  $\tilde{d}$  associated with  $s(\cdot)$  is

$$IV_{\tilde{d}}(f) = \frac{\sum_{\sigma \in \mathcal{M}_{\tilde{d}}^f} s_{\sigma}(f)}{\binom{d-1}{\tilde{d}-1}},$$

where  $\mathcal{M}_{\tilde{d}}^f = \{\sigma \in \mathcal{M}_{\tilde{d}} : f \in \sigma\}$  is the collection of subsets with size  $\tilde{d}$  that contains feature  $f$ , and  $s_{\sigma}(f)$  is the score assigned to feature  $f$  by computing  $s(\sigma; X)$ .

Intuitively, the inclusion value illustrates how much gain in score a feature  $f$  could provide on average, when it is included in the feature subset of size  $\tilde{d}$ . Our feature selection scheme is constructed based on inclusion value estimation, as summarized in Algorithm 1. We call it **Inclusion Value Feature Selection (IVFS)**. Roughly speaking, the algorithm selects features with highest estimated inclusion value, which is derived via  $k$  random sub-samplings of both features and observations. One benefit is that IVFS considers complicated feature interactions by evaluating subset of features together in each iteration.

**Special cases.** The IVFS scheme includes several popular methods as special cases based on different score function (i.e., the inclusion value).

- **Permutation importance.** For each feature in a random subset  $\mathcal{F}$ , if we set  $s_{\mathcal{F}}(f)$  as the difference between the performance (e.g., classification accuracy, regression mean squared error) using  $\mathcal{F}$  and the performance when feature vector  $f$  is randomly permuted, then IVFS becomes the feature selection algorithm via supervised random forest permutation importance (Strobl et al. 2008).
- **RKNN.** When we set  $s_{\mathcal{F}}(f) = -\mathcal{L}_{\mathcal{F}}, \forall f \in \mathcal{F}$ , where  $\mathcal{L}$  is the KNN classification error rate, IVFS becomes supervised RKNN (Li, Harner, and Adjeroh 2011).

Note that for random forest, the score function  $s_{\mathcal{F}}(f)$  is different for each feature  $f \in \mathcal{F}$ , while in RKNN, all the features in a random subset share a same score.

---

**Algorithm 1:** IVFS scheme for feature selection

---

**Input:** Data matrix  $X \in \mathbb{R}^{n \times d}$ ; Number of subsets  $k$ ;  
Number of features used for each subset  $\tilde{d}$ ;  
Number of samples for each subset  $\tilde{n}$ ;  
Target dimension  $d_0$

**Initialize:** Counters for each feature  $c_i = 0, i = 1, \dots, d$ ;  
Cumulative score for each feature  $S_i = 0$

```
1 for  $t = 1$  to  $k$  do
2   Randomly sample a size  $\tilde{d}$  feature set  $\mathcal{F} \in \mathcal{M}_{\tilde{d}}$ 
3   Randomly sub-sample  $X_{\mathcal{F}}^{sub} \in \mathbb{R}^{\tilde{n} \times \tilde{d}}$ , with  $\tilde{n}$ 
   observations and features in  $\mathcal{F}$ 
4   for  $f$  in  $\mathcal{F}$  do
5     Update counter  $c_f = c_f + 1$ 
6     Update score  $S_f = S_f + s_{\mathcal{F}}(f)$ 
7 Set  $S_i = \frac{S_i}{c_i}$  for  $i = 1, 2, \dots, d$ 
Output: Select top  $d_0$  features with highest score
```

---

### Analysis of IVFS

**Theorem 1** (Asymptotic  $k$ ). *Suppose  $k \rightarrow +\infty$ ,  $\tilde{n} = n$ ,  $s_{\sigma}(f)$  has finite variance  $\forall f \in \mathcal{I}$ , and the  $IV_{d_0}$  for different features are all distinct, then IVFS algorithm will select top  $d_0$  features with highest  $IV_{\tilde{d}}$  with probability 1.*

*Proof.* The algorithm is equivalent to finding the top  $d_0$  features with the largest

$$\hat{IV}_{\tilde{d}}(f) = \frac{\sum_{\sigma \in \hat{\mathcal{C}}_{\tilde{d}}^f} s_{\sigma}(f)}{|\hat{\mathcal{C}}_{\tilde{d}}^f|},$$

where  $\hat{\mathcal{C}}_{\tilde{d}}^f = \{\sigma \in \hat{\mathcal{C}}_{\tilde{d}} : f \in \sigma\}$ , and  $\hat{\mathcal{C}}_{\tilde{d}}$  is the collection of all  $k$  chosen random subsets. By central limit theorem, when  $k \rightarrow \infty$  we have for some  $\tau$  and  $\forall f$ ,

$$\frac{\hat{IV}_{\tilde{d}}(f) - IV_{\tilde{d}}(f)}{\sqrt{k}} \rightarrow N(0, \tau^2).$$

Let  $\delta$  be the difference between the  $d_0$ -highest  $IV_{\tilde{d}}$  and the  $(d_0 + 1)$  highest, then for  $\forall f$ , for any  $\epsilon > 0$  there exists a  $K$  such that when  $k > K$ , the probability of  $|\hat{IV}_{\tilde{d}}(f) - IV_{\tilde{d}}(f)| > \delta/2$  is less than  $\epsilon$ . Taking  $\epsilon \rightarrow 0$  and  $k \rightarrow \infty$ , the theorem is proved.  $\square$

In the following, we look at the case where  $s_{\mathcal{F}}(f)$  are equal for every  $f \in \mathcal{F}$  (e.g., RKNN). As a result, we may re-write the score function as  $s(\mathcal{F})$  evaluated on subsets. The next assumption on *monotonicity* appears commonly in feature selection literature, similar in spirit to (Narendra and Fukunaga 1977; Foroutan and Sklansky 1987), etc.

**Assumption 1** (Monotonicity). *There exists a  $\tilde{d}$ -dimensional set  $\Omega \in \mathcal{M}_{\tilde{d}}$  such that  $\forall \mathcal{F}, \mathcal{F}' \in \mathcal{M}_{\tilde{d}}$ , if  $(\mathcal{F} \cap \Omega) \subseteq (\mathcal{F}' \cap \Omega)$ , then  $s(\mathcal{F}) \leq s(\mathcal{F}')$ .*

Basically, this assumption says that there is a subset  $\Omega$  of “dominant features”: For any two subsets  $\mathcal{F}, \mathcal{F}'$  with same size, if  $(\mathcal{F} \cap \Omega)$  is contained in  $(\mathcal{F}' \cap \Omega)$ , then the score of  $\mathcal{F}'$  is no smaller than that of  $\mathcal{F}$ . It turns out that this dominant

set is indeed optimal, and it is also the solution that IVFS converges to in large  $k$  limit.

**Theorem 2** (Optimality). *Under Assumption 1, we have*

$$\Omega = \arg \max_{\mathcal{F} \in \mathcal{M}_{\tilde{d}}} s(\mathcal{F}),$$

and  $\Omega$  is the set of  $\tilde{d}$  features with the highest  $IV_{\tilde{d}}$ .

*Proof.* It suffices to show that for  $\forall f \in \Omega, g \notin \Omega, IV_{\tilde{d}}(f) \geq IV_{\tilde{d}}(g)$ . We have by assumption

$$\begin{aligned} & IV_{\tilde{d}}(f) - IV_{\tilde{d}}(g) \\ &= \frac{1}{\tilde{d}} \left( \sum_{\substack{\mathcal{F} \in \mathcal{M}_{\tilde{d}-1} \\ f \notin \mathcal{F}}} s(\mathcal{F} \cup \{f\}) - \sum_{\substack{\mathcal{F} \in \mathcal{M}_{\tilde{d}-1} \\ g \notin \mathcal{F}}} s(\mathcal{F} \cup \{g\}) \right) \\ &= \frac{1}{\tilde{d}} \sum_{\substack{\mathcal{F} \in \mathcal{M}_{\tilde{d}-1} \\ f \notin \mathcal{F}, g \notin \mathcal{F}}} (s(\mathcal{F} \cup \{f\}) - s(\mathcal{F} \cup \{g\})) \geq 0, \end{aligned}$$

which proves the second argument. Since for  $\forall \mathcal{F} \in \mathcal{M}_{\tilde{d}}, (\mathcal{F} \cap \Omega) \subseteq (\Omega \cap \Omega)$ , we have  $\Omega = \arg \max_{\mathcal{F} \in \mathcal{M}_{\tilde{d}}} s(\mathcal{F})$ .  $\square$

Together with Theorem 1, we know that if we set  $\tilde{d} = d_0$ , under Assumption 1, IVFS would converge to the minimal score feature set. For instance, in the case of RKNN, the selected features would minimize the KNN error rate.

**Choosing  $\tilde{d}$ .** In practice (when  $k \ll \binom{d}{d_0}$ ), we are actually drawing random samples from the population, and use feature scores as estimation of true inclusion values. An interesting fact is that, IVFS actually uses  $\hat{IV}_{\tilde{d}}$  to estimate  $IV_{d_0}$ . This makes sense since 1) we expect features with high  $IV_{\tilde{d}}$  to have high  $IV_{d_0}$  as well, and 2) we care more about the rank of feature scores rather than the exact values. Hence, setting  $\tilde{d} > d_0$  may not defect the model performance.

**Choosing  $\tilde{n}$ .** We also have a parameter  $\tilde{n}$  which controls the number of random observations for each subset. A relatively small  $\tilde{n}$  is extremely helpful to accelerate the algorithm for scalable implementation on large datasets. In existing methods, however, sub-sampling is not commonly used.

- In the original proposal of random forest (Breiman 2001), the authors suggested to set  $\tilde{n} = n$  with replacement. Later on, a popular variant (Geurts, Ernst, and Wehenkel 2006) chose to disable sub-sampling procedure. In general, in all variants of random forest, sub-sampling is not recommended (Tang, Garreau, and von Luxburg 2018).
- In RKNN (Li, Harner, and Adjeroh 2011), the author did not consider sub-sampling training points, either.

For supervised learning, the above phenomenon seems reasonable, since sub-sampling the training data may harm the learning capacity of each random subset, especially with high dimensions. However, as shall be seen from next sections, when applying IVFS for the purpose of unsupervised topology preservation, we can choose a very small  $\tilde{n}$  without losing much capacity. This makes IVFS a strong candidate for dealing with large scale datasets. In general, a good choice of  $\tilde{n}$  depends on the specific problem.

**Advantages of IVFS.** On a high level, IVFS has the following nice features:

- Intuitive formulation, no complicated computation.
- IVFS can well handle the problems where the optimization problem is very hard to solve (or intractable), as long as the loss function can be computed efficiently.
- IVFS can be applied to large datasets efficiently by using a small  $\tilde{n}$ , which is feasible for some applications.

In the following sections, we design a topology preserving feature selection algorithm by combining IVFS framework and ideas from topological data analysis (TDA).

### Preliminaries on Computational Topology

In this section, we provide some intuition to several important concepts in computational topology. Interested readers are referred to (Edelsbrunner and Harer 2010) for more detailed introduction. A  $p$ -dimensional simplex is defined as

$$\gamma_p = \{\theta_0 x_0 + \dots + \theta_p x_p \mid \theta_i > 0 \forall i, \sum_{i=0}^p \theta_i = 1\},$$

where  $x_0, \dots, x_p$  are affinely independent points in  $\mathbb{R}^p$ . For instance, a 1-simplex is a line segment, and a 2-simplex is a triangle, etc. A *simplicial complex*  $\mathcal{C}$  is then formed by gluing simplices in different dimensions together. In Euclidean space, the most commonly used complex is the *Vietoris-Rips complex*, with an example given in Figure 1. It is formed by connecting points with distance smaller than a given threshold  $\alpha$ . If we gradually increase  $\alpha$  from 0 to  $\infty$ , the number of edges will increase from 0 to  $n^2$  eventually. The distance associated with each edge, is called the *filtration* for Rips complex. As  $\alpha$  increases, topological features with different dimension (e.g., 0 for connected components, 1 for loops, 2 for voids, etc.) will appear and disappear. We call the pair of birth and death time (the  $\alpha$  value) of a  $p$ -dimensional topological feature as a  $p$ -dimensional *persistent barcode*. The  $p$ -dimensional *persistent diagram* is a multiset of all these barcodes. An example persistent diagram is plotted in Figure 1 right panel. Note that we can always normalize the filtration function to be bounded in  $[0, 1]$ . Often, barcodes with length less than a small number  $\epsilon$  are regarded as noise and eliminated from the diagram. In many applications, useful features are then retrieved from persistent diagrams (e.g., persistent image (Adams et al. 2017) and persistent landscape (Bubenik 2015)) as inputs fed into learning machines.

### Topology Preservation via IVFS

In most applications of TDA, the first step is to generate a persistent diagram summarizing the topological patterns. When feature selection is adopted in TDA, the persistent diagram should be accurately preserved. In this section, we propose a variant under IVFS scheme that achieves this goal.

#### Distance measure

In our study, we focus on Rips complex which is widely used for real-valued data. The filtration of Rips complex is based on distances between data points (cf. Figure 1). In this

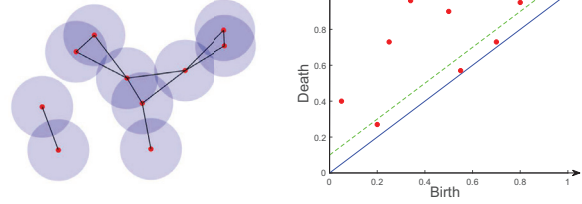


Figure 1: Left panel: an example of Rips complex. Points that are close are connected. Dots, lines and triangles forms 0, 1 and 2-dimensional complex respectively. Right panel: an example of persistent diagram. Each red point represents a (birth, death) time of a topological feature. The green line is a threshold: only barcodes that are stable (points above the green line) are included in the final barcode set.

paper, we will mainly focus on Euclidean distance. Consider a distance matrix  $(D_{ij})$ , with the  $(i, j)$ -th entry defined as  $D_{ij} = \|x_i - x_j\|_2$ , where  $x_i$  and  $x_j$  are two sample points, and  $\|\cdot\|_2$  is the  $l_2$  norm for vectors. We divide  $D$  by its largest entry to normalize all distances to  $[0, 1]$ . Other similarities such as cosine and generalized min-max (GMM) (Li 2017; Li and Zhang 2017) can also be adopted.

#### Distances between persistent diagrams

Recall that our objective involves minimizing the difference between persistent diagrams. The following two distances measures between diagrams are widely used in TDA.

**Definition 2.** For two persistent diagrams  $\Psi$  and  $\Gamma$ , define *Wasserstein distance* ( $w_p^q$ ) and *Bottleneck distance* ( $w_\infty$ )

$$w_p^q(\Psi, \Gamma) = \inf_{\phi} \left( \sum_{x \in \Psi} \|x - \phi(x)\|_q^p \right)^{\frac{1}{p}},$$

$$w_\infty(\Psi, \Gamma) = \inf_{\phi} \sup_{x \in \Psi} \|\phi(x)\|_\infty,$$

where  $\phi$  is taken over all bijections  $\Psi \rightarrow \Gamma$  and  $p, q \in \mathbb{N}$ .

#### Objective function

Now we are ready to formally state our objective function. Recall the notations  $\mathcal{I} = \{1, 2, \dots, d\}$ , and  $\mathcal{M}_{\vec{d}} = \{\sigma \in \mathcal{I}^{\vec{d}} : \sigma_i \neq \sigma_j \text{ for } \forall i \neq j\}$ . To achieve topology preserving feature selection, we minimize following loss function,

$$\min_{\mathcal{F} \in \mathcal{M}_{d_0}} w_*(\mathcal{D}(X), \mathcal{D}(X_{\mathcal{F}})), \quad (3)$$

where  $w_*$  denotes Wasserstein or bottleneck distance. This way, we find the subset that best preserves the topological signatures of original data. However, the mapping between feature space and persistent diagram is so sophisticated that analytical approach to (3) is hard to derive. This is exactly the circumstance where IVFS is particularly effective.

Note that the computational cost to generate persistent diagram is non-negligible even for data of moderate size. Hence, directly applying IVFS with objective loss (3) would

be extremely slow. To this end, we leverage from the stability property of persistent diagrams to propose an alternative solution. We re-state the theorems as below.

**Theorem 3.** (Chazal, De Silva, and Oudot 2014) Suppose  $X$  is the point set,  $f$  and  $f'$  are two Lipschitz functions. Let  $\mathcal{D}(X, f)$  and  $\mathcal{D}(X, f')$  denote the persistent diagram built upon  $X$  using filtration function  $f$  and  $f'$ , respectively. Under some technical conditions, the bottleneck and Wasserstein distances are bounded by

$$w_\infty(\mathcal{D}(X, f), \mathcal{D}(X, f')) \leq C_1 \|f - f'\|_\infty,$$

$$w_p^w(\mathcal{D}(X, f), \mathcal{D}(X, f')) \leq C_2 \|f - f'\|_\infty^{C_3},$$

where  $C_1$ ,  $C_2$  and  $C_3$  are a universal constant independent of  $X$ , and  $\|\cdot\|_\infty$  refers to the infinite norm.

In words, the theorem says that when we change filtration from  $f$  to  $f'$ , the change in persistent diagrams would be bounded linearly in the  $\|f - f'\|_\infty$  for Bottleneck distance, and polynomially for Wasserstein distance. Since the filtration for Rips complex is the pairwise distances, we can alternatively control the  $l_\infty$  norm of the difference between two distance matrices. Thus, we substitute our objective to

$$\min_{\mathcal{F} \in \mathcal{M}_{d_0}} \|D_{\mathcal{F}} - D\|_\infty, \quad (4)$$

where  $\|A\|_\infty = \max_{i,j} |A_{ij}|$ ,  $D$  and  $D_{\mathcal{F}}$  are the distance matrix before and after feature selection. By (4), we get rid of the expensive computation of persistent diagrams, making the algorithm applicable to real world applications. Nevertheless, optimization regarding  $l_\infty$  is still non-trivial.

### IVFS- $l_\infty$ Algorithm and Extensions

The IVFS scheme (Algorithm 1) can be directly adapted to the topology preservation problem (4). At line 6, we substitute score function as

$$s_{\mathcal{F}}(f) = -\|D_{\mathcal{F}} - D\|_\infty, \quad \forall f \in \mathcal{F}, \quad (5)$$

and all other steps remain the same. This is called the **IVFS- $l_\infty$**  algorithm.

**Extensions.** We can easily extend this algorithm to other reasonable loss functions. Denote  $\|D - D_{\mathcal{F}}\|_1 = \sum_{i,j} |D_{ij} - D_{\mathcal{F}ij}|$  and  $\|D - D_{\mathcal{F}}\|_2$  the matrix Frobenius norm. One should expect that minimizing these two norms of  $(D - D_{\mathcal{F}})$  to be good alternatives, because small  $\|D - D_{\mathcal{F}}\|_1$  or  $\|D - D_{\mathcal{F}}\|_2$  is likely to result in a small  $\|D_{\mathcal{F}} - D\|_\infty$  as well. We will call these two methods **IVFS- $l_1$**  and **IVFS- $l_2$**  algorithm respectively. Both extensions are implemented simply by changing the lines in Algorithm 1 with corresponding loss and score function.

## Experiments

Following many previous works on feature selection (e.g. Zhao and Liu 2007; Liu et al. 2014; Li et al. 2018b)), we carry out extensive experiments on popular high-dimensional datasets from UCI repository (Asuncion and Newman 2007) and ASU feature selection database (Li et al. 2018a). The summary statistics are provided in Table 1. For all datasets, the features are standardized to mean zero and unit variance, and at most 300 features are selected.

## Methods and tuning parameters

We compare several popular similarity preserving methods:

- **SPEC:** We use the second type algorithm which performs the best, according to (Zhao and Liu 2007).
- **MCFS:** As guided in (Cai et al., 2010), we run MCFS with number of clusters  $M = \{5, 10, 20, 30\}$ .
- **GLSPFS:** It is a embedded method that learns to approximate the sample similarity matrix and will serve as a major baseline. Following (Liu et al. 2014), we chose the parameter combinations of  $\mu = \{0, 10^{-1:3}\}$ ,  $\lambda = \{10^{-1:3}\}$ .
- **IVFS- $l_\infty$**  and variants: We try following combinations:  $\tilde{d} = \{0.1 : 0.1 : 0.5\} \times d$ ,  $\tilde{n} = \{100, 0.1n, 0.3n, 0.5n\}$ . We run experiments with  $k = 1000, 3000, 5000$ . All the reported results are averaged over 5 repetitions.

## Evaluation Metrics

We compare each method by various widely adopted metrics that can well evaluate the quality of selected feature set.<sup>2</sup>

**KNN accuracy.** Following (Zhao and Liu 2007; Liu et al. 2014), etc., we test local structure preservation by KNN classification. Each dataset is randomly split into 80% training sets and 20% test set, on which the test accuracy is computed. We repeat the procedure 10 times and take the average accuracy. For number of neighbors, we adopt  $K \in \{1, 3, 5, 10\}$  and report the highest mean accuracy.

**Distances between persistent diagrams.** For  $X$  and  $X_{\mathcal{F}}$ , we compute the 1-dimensional persistent diagram  $\mathcal{D}$  and  $\mathcal{D}_{\mathcal{F}}$  with  $\alpha = 0.5$  and drop all barcodes with existing time less than  $\epsilon = 0.1$ . Wasserstein ( $w_1^1$ ) and Bottleneck ( $w_\infty$ ) distances are computed between the diagrams.

**Norms between distance matrix.** When the purpose of feature selection is to preserve the sample distance (or similarity), one straightforward measure should be the change between distance matrices  $D$  and  $D_{\mathcal{F}}$ . We compute  $\|D - D_{\mathcal{F}}\|_1$ ,  $\|D - D_{\mathcal{F}}\|_2$  and  $\|D - D_{\mathcal{F}}\|_\infty$  to evaluate the closeness of these two matrices.

**Running time.** We compare the running time for a single run, with fixed parameter setting: **MCFS:**  $k = 10$ , **GLSPFS:**  $\mu = 1$ ,  $\lambda = 1$ , and **IVFS:**  $k = 1000$ ,  $\tilde{n} = 0.1n$ ,  $\tilde{d} = 0.3d$ .

## Results

**Overall performance.** Table 1 summarizes the results. All the datasets are high-dimensional, and *Isolet*, *RELATHE*, *COIL20* have relatively larger size with around 1500 samples. For algorithms with tuning parameters, we report the best result among all parameters and number of selected features for each metric. From Table 1, we observe:

- IVFS- $l_\infty$  provides smallest  $w_1^1$ , and  $w_\infty$  on almost all datasets. Moreover, the  $l_1$ ,  $l_2$  and  $l_\infty$  norms are significantly reduced on all the data—The distance and topology preserving capability is essentially improved.
- On all datasets, IVFS- $l_\infty$  also beats other methods in terms of KNN accuracy, which indicates its superiority on supervised tasks and local manifold preservation.

<sup>2</sup>We also tested normalized mutual information (NMI) in the experiment. The pattern was very similar to KNN accuracy.

Table 1: Experiments on high dimensional data using normalized Euclidean distance. # n is the number of samples, # d is the dimensionality and # C is the number of classes. The unit of  $L_1/n^2$  is ( $\times 10^{-2}$ ). If  $n < 1000$ ,  $\tilde{n} = 0.1n$ ; otherwise,  $\tilde{n} = 100$ .

Dataset	#n	#d	#C	Methods	KNN	$w_1^1$	$w_\infty$	$L_\infty$	$L_1/n^2$	$L_2$	Time (Sec)
CLL-SUB-111	111	11340	3	SPEC	58.2%	2.35	0.05	0.42	9.49	13.86	0.85
				MCFS	63.9%	2.91	0.04	0.21	4.19	5.85	3.35
				GLSPFS	63.9%	2.44	<b>0.02</b>	0.30	6.89	9.58	11.29
				IVFS- $l_\infty 1000$	<b>68.7%</b>	<b>2.09</b>	<b>0.02</b>	<b>0.11</b>	<b>2.81</b>	<b>3.94</b>	5.27
Lymphoma	96	4026	9	SPEC	81.5%	0.31	0.05	0.25	6.38	7.64	0.28
				MCFS	92.5%	0.09	0.02	0.14	2.46	2.95	1.51
				GLSPFS	92.0%	0.12	0.03	0.19	4.22	5.10	1.50
				IVFS- $l_\infty 1000$	<b>94.0%</b>	<b>0.06</b>	<b>0.01</b>	<b>0.08</b>	<b>1.90</b>	<b>2.30</b>	1.28
Orlraws10P	100	10304	10	SPEC	81.5%	0.92	0.03	0.49	13.21	16.51	0.71
				MCFS	92.0%	0.50	0.03	0.20	4.63	5.90	2.69
				GLSPFS	93.5%	0.66	<b>0.02</b>	0.24	5.73	7.13	12.49
				IVFS- $l_\infty 1000$	<b>98.0%</b>	<b>0.47</b>	<b>0.02</b>	<b>0.08</b>	<b>1.85</b>	<b>2.35</b>	4.08
Pixrow10P	100	10000	10	SPEC	98.0%	2.43	0.07	0.57	13.81	17.29	0.68
				MCFS	99.0%	1.76	0.05	0.29	6.70	8.43	2.64
				GLSPFS	99.0%	1.41	0.05	0.26	6.48	8.02	14.68
				IVFS- $l_\infty 1000$	<b>100%</b>	<b>0.60</b>	<b>0.03</b>	<b>0.07</b>	<b>2.03</b>	<b>2.50</b>	3.71
Prostate-GE	102	5966	2	SPEC	73.3%	3.70	0.09	0.38	12.39	15.37	0.43
				MCFS	81.9%	0.94	0.05	0.22	5.06	6.27	2.01
				GLSPFS	83.8%	0.64	0.03	0.27	4.61	7.39	4.45
				IVFS- $l_\infty 1000$	<b>87.6%</b>	<b>0.40</b>	<b>0.02</b>	<b>0.06</b>	<b>1.44</b>	<b>1.96</b>	1.89
SMK-CAN-187	187	19993	2	SPEC	70.3%	4.10	0.06	0.59	10.28	25.26	2.13
				MCFS	66.1%	0.99	0.03	0.29	3.82	9.91	7.06
				GLSPFS	71.1%	1.57	0.04	0.26	5.81	14.35	55.90
				IVFS- $l_\infty 1000$	<b>72.6%</b>	<b>0.58</b>	<b>0.02</b>	<b>0.09</b>	<b>2.00</b>	<b>3.86</b>	9.62
WarpPIE10P	130	2400	10	SPEC	85.5%	1.07	0.06	0.44	21.72	22.31	1.30
				MCFS	95.7%	0.60	0.05	0.14	5.90	6.14	1.49
				GLSPFS	90.0%	0.68	0.05	0.16	7.10	7.39	2.03
				IVFS- $l_\infty 1000$	<b>95.9%</b>	<b>0.50</b>	<b>0.04</b>	<b>0.05</b>	<b>2.60</b>	<b>3.04</b>	1.11
COIL20	1440	1024	20	SPEC	98.2%	16.07	0.07	0.35	16.96	258.5	17.02
				MCFS	99.6%	3.50	0.08	0.30	2.63	42.43	3.92
				GLSPFS	99.9%	<b>1.78</b>	0.06	0.20	2.02	40.60	8.98
				IVFS- $l_\infty 1000$	<b>100%</b>	2.16	<b>0.05</b>	<b>0.18</b>	<b>1.41</b>	<b>25.66</b>	3.86
Isolet	1560	617	26	SPEC	81.6%	63.85	0.08	0.43	11.96	209.9	12.86
				MCFS	82.3%	20.18	0.04	0.18	2.47	52.14	2.20
				GLSPFS	88.2%	11.64	0.03	0.13	1.94	38.26	7.34
				IVFS- $l_\infty 1000$	<b>88.7%</b>	<b>2.09</b>	<b>0.01</b>	<b>0.08</b>	<b>1.37</b>	<b>27.04</b>	2.78
RELATHE	1427	4322	2	SPEC	72.8%	36.04	0.10	0.83	15.08	263.6	69.20
				MCFS	68.8%	9.68	0.08	0.35	8.43	142.6	9.34
				GLSPFS	72.5%	7.32	0.07	0.32	5.21	94.32	30.46
				IVFS- $l_\infty 1000$	<b>75.6%</b>	<b>0.70</b>	<b>0.05</b>	<b>0.24</b>	<b>1.90</b>	<b>40.90</b>	8.87

**Robustness.** We plot  $\|D - D_{\mathcal{F}}\|_2$  and  $w_\infty(\mathcal{D}, \mathcal{D}_{\mathcal{F}})$  against the number of selected features in Figure 2 and Figure 3, respectively. SPEC performs very poorly on high-dimensional datasets. We observe clearly a trend that IVFS keeps lifting its performance as number of features increases. This robustness comes from the fact that the inclusion value intrinsically contains rich information about features interactions.

**Stability.** We bootstrap samples from original dataset to mimic the process of sampling from population. Denote  $\mathcal{F}_X$  and  $\mathcal{F}_B$  the subset chosen based on original data and bootstrap data respectively. We count  $|G|$  with  $G = \{f : f \in \mathcal{F}_X, f \notin \mathcal{F}_B\}$ . We use same parameters as for testing the

running time. The results are averaged over 5 repetitions. In principle, the selected feature pool should not vary significantly if we only change a few samples (by bootstrap), since the “truly” important features should be independent of the samples. In Table 2, we see that IVFS- $l_\infty$  is hardly affected by the bootstrapping process, while other methods are much more sensitive and gives very different solutions.

**Efficiency-capacity trade-off.** From Table 1, we also see that IVFS- $l_\infty$  is comparable with MCFS in terms of running time, and is more efficient than GLSPFS. In particular, GLSPFS may run very slowly when it takes a long time to converge (e.g., *SMK-CAN-187*). Additionally, for datasets

Table 2: Stability under bootstrap: the number of different selected features between original data and bootstrap data.

	SPEC	MCFS	GLSPFS	IVFS- $l_\infty$ 1000
CLL-SUB-111	187.2	271.8	243.6	<b>12.2</b>
Lymphoma	156.6	260.8	260.4	<b>4.4</b>
Orlraws10P	119.2	283.6	267.2	<b>7.8</b>
Pixrow10P	143.0	284.8	266.0	<b>6.8</b>
Prostate-GE	125.8	264.4	236.2	<b>12.0</b>
SMK-CAN-187	125.8	264.4	236.2	<b>18.6</b>
WarpPIE10P	56.0	241.6	191.0	<b>4.8</b>
COIL20	14.0	178.2	115.4	<b>5.6</b>
Isolet	12.8	124.5	80.0	<b>3.3</b>
RELATHE	21.4	145.0	102.8	<b>8.4</b>

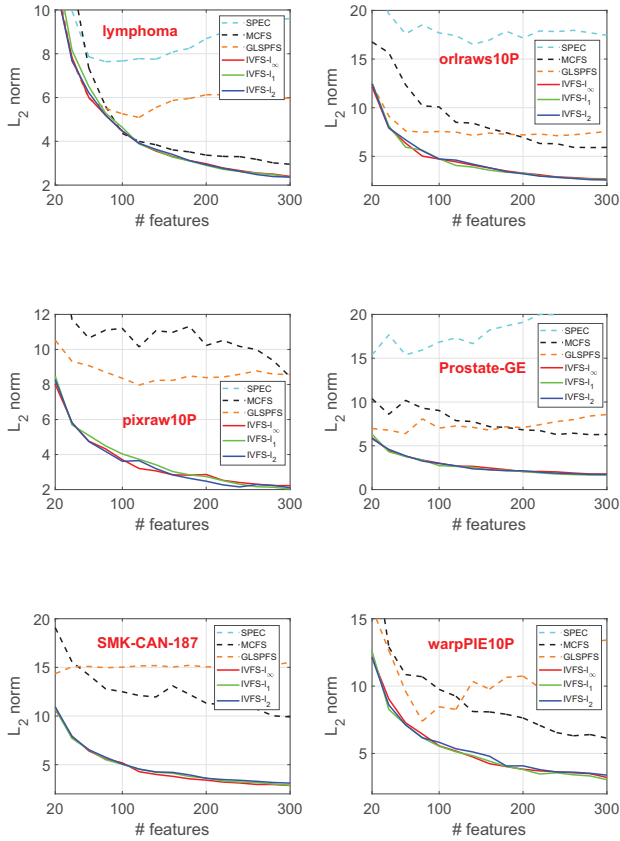


Figure 2:  $l_2$  norm vs. number of features.

with large size, SPEC and GLSPFS gets slower due to large matrix inverse and singular value decomposition.

The computational cost of our IVFS algorithms depends tightly on the sub-sampling rate  $\tilde{n}/n$  and number of random subsets  $k$ . As one would expect, there exists a trade-off between computational efficiency and distance preserving power. In Figure 4, we plot the relative performance (set the value for  $k = 5000$ ,  $\tilde{n} = 0.5n$  as 1) of different  $k$  and  $\tilde{n}$ , averaged over all datasets and number of chosen features. In general, the performance of IVFS boosts as  $k$  and  $\tilde{n}$  increase

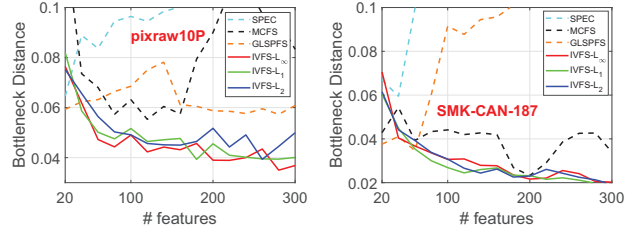


Figure 3: Bottleneck distance vs. number of features.

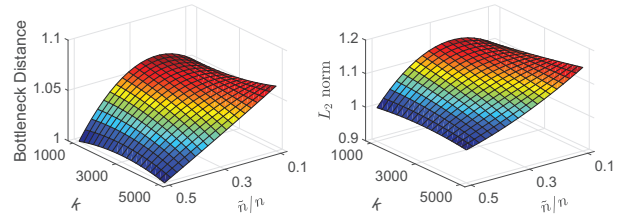


Figure 4: Performance comparison across different number of subsets  $k$  and sub-sampling ratio  $\tilde{n}/n$ .

because of more accurate estimate of the inclusion values.

Note that in our experiment, when the sample size is relatively large (i.e., greater than 1000), we simply fix  $\tilde{n} = 100$ , and IVFS still outperforms other methods (on *COIL20*, *RELATHE*). Thus, to achieve better efficiency, we recommend practitioners to set  $k = 1000$  as default. For  $\tilde{n}$ , if the data size is not very large, we suggest using  $\tilde{n} = 0.1n$  as the first try. Otherwise, one may threshold  $\tilde{n}$  at a small number (e.g., 100 in our experiment). This way, the running time, when fixing  $k$  and  $\tilde{d}$ , becomes approximately constant.

## Discussion

For supervised random forest, it is conventional to set as default  $\tilde{d} = \sqrt{d}$ , which is around  $10^2$  when  $d$  is around  $10^4$ . However, in this case we have to use large  $k$  to make sure that every feature is evaluated for not too few times. We observe that there is not much gain in performance with small  $\tilde{d}$  and large  $k$  combination. Thus, we suggest to use a relatively large  $\tilde{d}$  to speed up the algorithm for higher efficiency.

It is worth mentioning that there are also hybrid strategies for feature selection, where one first determines a pool of features, and then select ultimate feature set from the pool through another round of screening. One thing we observe from the experiments is that IVFS is much more stable and robust than other algorithms, which means that the features selected are mostly “good” features that help with reducing the loss. Therefore, IVFS is also suitable to be applied in the pool selection procedure for such hybrid methods.

## Conclusion

In this paper, we propose IVFS, a unified feature selection scheme based on random subset methods. After we show its connection with existing methods such as random forest and RKNN, we propose IVFS- $l_\infty$  and several variants that can preserve the pairwise distance and topological signatures (persistent diagram) of the original dataset more precisely than the competing similarity preserving algorithms. This would be very helpful for applications requiring high-level distance preservation, e.g. topological data analysis. In the experiments, we evaluate the distance preserving capability of different algorithms through to demonstrate the effectiveness of the proposed IVFS algorithms. We also demonstrate that a sharp sub-sampling rate can be effectively adopted on this problem to speed up the algorithm on large datasets.

## References

- Adams, H.; Emerson, T.; Kirby, M.; Neville, R.; Peterson, C.; Shipman, P. D.; Chepushtanova, S.; Hanson, E. M.; Motta, F. C.; and Ziegelmeier, L. 2017. Persistence images: A stable vector representation of persistent homology. *J. Mach. Learn. Res.* 18:8:1–8:35.
- Asuncion, A., and Newman, D. 2007. Uci machine learning repository.
- Breiman, L. 2001. Random forests. *Machine learning* 45(1):5–32.
- Bubenik, P. 2015. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.* 16(1):77–102.
- Cai, D.; Zhang, C.; and He, X. 2010. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 333–342.
- Chazal, F.; De Silva, V.; and Oudot, S. 2014. Persistence stability for geometric complexes. *Geometriae Dedicata* 173(1):193–214.
- Du, L., and Shen, Y. 2015. Unsupervised feature selection with adaptive structure learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 209–218.
- Edelsbrunner, H., and Harer, J. 2010. *Computational topology: an introduction*. American Mathematical Soc.
- Foroutan, I., and Sklansky, J. 1987. Feature selection for automatic classification of non-gaussian data. *IEEE Transactions on Systems, Man, and Cybernetics* 17(2):187–198.
- Geurts, P.; Ernst, D.; and Wehenkel, L. 2006. Extremely randomized trees. *Machine Learning* 63(1):3–42.
- Guo, J., and Zhu, W. 2018. Dependence guided unsupervised feature selection. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2232–2239.
- Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3(Mar):1157–1182.
- He, X.; Cai, D.; and Niyogi, P. 2006. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems (NIPS)*, 507–514.
- Hofer, C.; Kwitt, R.; Niethammer, M.; and Uhl, A. 2017. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems (NIPS)*, 1634–1644.
- Li, P., and Zhang, C.-H. 2017. Theory of the GMM kernel. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, 1053–1062.
- Li, Z.; Yang, Y.; Liu, J.; Zhou, X.; and Lu, H. 2012. Unsupervised feature selection using nonnegative spectral analysis. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*.
- Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; and Liu, H. 2018a. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)* 50(6):94.
- Li, J.; Wu, L.; Dani, H.; and Liu, H. 2018b. Unsupervised personalized feature selection. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 3514–3521.
- Li, S.; Harner, E. J.; and Adjeroh, D. A. 2011. Random knn feature selection—a fast and stable alternative to random forests. *BMC bioinformatics* 12(1):450.
- Li, P. 2017. Linearized GMM kernels and normalized random Fourier features. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 315–324.
- Liu, X.; Wang, L.; Zhang, J.; Yin, J.; and Liu, H. 2014. Global and local structure preservation for feature selection. *IEEE Transactions on Neural Networks and Learning Systems* 25(6):1083–1095.
- Narendra, P. M., and Fukunaga, K. 1977. A branch and bound algorithm for feature subset selection. *IEEE Transactions on computers* (9):917–922.
- Nie, F.; Zhu, W.; and Li, X. 2016. Unsupervised feature selection with structured graph optimization. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, 1302–1308.
- Peng, H.; Long, F.; and Ding, C. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence* 27(8):1226–1238.
- Qian, M., and Zhai, C. 2013. Robust unsupervised feature selection. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 1621–1627.
- Räsänen, O., and Pohjalainen, J. 2013. Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 210–214.
- Strobl, C.; Boulesteix, A.-L.; Kneib, T.; Augustin, T.; and Zeileis, A. 2008. Conditional variable importance for random forests. *BMC bioinformatics* 9(1):307.
- Tang, C.; Garreau, D.; and von Luxburg, U. 2018. When do random forests fail? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2987–2997.
- Yu, L., and Liu, H. 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, 856–863.
- Zhao, Z., and Liu, H. 2007. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the Twenty-Fourth international conference on Machine learning (ICML)*, 1151–1157.
- Zhao, Z.; Wang, L.; Liu, H.; and Ye, J. 2013. On similarity preserving feature selection. *IEEE Transactions on Knowledge and Data Engineering* 25(3):619–632.
- Zhu, X.; Zhu, Y.; Zhang, S.; Hu, R.; and He, W. 2017. Adaptive hypergraph learning for unsupervised feature selection. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, 3581–3587.