Ultra High-Dimensional Nonlinear Feature Selection for Big Biological Data

Makoto Yamada, *Member, IEEE*, Jiliang Tang, Jose Lugo-Martinez, Ermin Hodzic, Raunak Shrestha, Avishek Saha, Hua Ouyang, Dawei Yin, Hiroshi Mamitsuka, Cenk Sahinalp, Predrag Radivojac, Filippo Menczer, and Yi Chang

Abstract—Machine learning methods are used to discover complex nonlinear relationships in biological and medical data. However, sophisticated learning models are computationally unfeasible for data with millions of features. Here, we introduce the first feature selection method for nonlinear learning problems that can scale up to large, ultra-high dimensional biological data. More specifically, we scale up the novel Hilbert-Schmidt Independence Criterion Lasso (HSIC Lasso) to handle millions of features with tens of thousand samples. The proposed method is guaranteed to find an optimal subset of maximally predictive features with minimal redundancy, yielding higher predictive power and improved interpretability. Its effectiveness is demonstrated through applications to classify phenotypes based on module expression in human prostate cancer patients and to detect enzymes among protein structures. We achieve high accuracy with as few as 20 out of one million features—a dimensionality reduction of 99.998 percent. Our algorithm can be implemented on commodity cloud computing platforms. The dramatic reduction of features may lead to the ubiquitous deployment of sophisticated prediction models in mobile health care applications.

Index Terms—Feature selection, kernel methods, Hilbert-Schmidt independence criterion, biomarker discovery

1 Introduction

Life sciences are going through a revolution thanks to the possibility to collect and learn from massive biological data [1]. Efficient processing of such "big data" is extremely important for many medical and biological applications, including disease classification, biomarker discovery, and drug development. The complexity of biological data is dramatically increasing due to improvements in measuring devices such as next-generation sequencers, microarrays and mass spectrometers [2]. As a result, we must deal with data that includes many observations (hundreds to tens of thousands) and even larger numbers of features (thousands to millions). Machine learning algorithms are charged with learning patterns and extracting actionable information from biological data. These techniques have been used successfully in various analytical tasks, such as genome-wide association studies [3] and gene selection [4].

However, the scale and complexity of big biological data pose new challenges to existing machine learning algorithms. There is a trade-off between scalability and complexity: linear methods scale better to large data, but cannot model complex patterns. Nonlinear models can handle complex relationships in the data but are not scalable to the size of current datasets. In particular, learning nonlinear models requires a number of observations that grows exponentially with the number of features [4], [5]. Biological data generated by modern technology has as many as millions of features, making the learning of nonlinear models unfeasible with existing techniques. To make matters worse, current nonlinear approaches cannot take advantage of distributed computing platforms.

A promising approach to make nonlinear analysis of big biological data computationally tractable is to reduce the number of features. This method is called *feature selection* [4]. Biological data is often represented by matrices where rows denote features and columns denote observations. Feature selection aims to identify a subset of features (rows)

- *M. Yamada is with the Center for Advanced Intelligence Project, RIKEN, Chuo-ku 103-0027, Japan. E-mail: makoto.yamada@riken.jp.*
- *J. Tang is with the Department of Computer Science, Arizona State Univerity, Tempe, AZ 85282. E-mail: Jiliang.Tang@asu.edu.*
- *J. Lugo-Martinez is with the Department of Computer Science and Informatics, Indiana University Bloomington, Bloomington, IN 47405. E-mail: jlugomar@indiana.edu.*
- *E. Hodzic is with the Department of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada. E-mail: ehodzic@sfu.ca.*
- *R. Shrestha is with the Department of Bioinformatics, University of British Columbia, Vancouver, BC V6T 1Z4, Canada. E-mail: rshrestha@prostatecentre.com.*
- *A. Saha is with A9, Amazon, 130 Lytton Ave Ste 300 Palo Alto, CA 94301. E-mail: avishek.saha@gmail.com.*
- *H. Ouyang is with Apple, 1 Infinite Loop Cupertino, CA 95014. E-mail: huaouyang@gmail.com.*
- *D. Yin is with JD.com, 8 Beichen West Rd. Beijing, China, 10000. E-mail: yindawei@acm.org.*
- *H. Mamitsuka is with the Institute for Chemical Research, Kyoto University, Uji 611-0011, Japan. E-mail: mami@kuicr.kyoto-u.ac.jp.*
- *C. Sahinalp is with the Department of Computer Science, Indiana University Bloomington, Bloomington, IN 47405. E-mail: cenksahi@indiana.edu.*
- *P. Radivojac and F. Menczer are with the School of Informatics, Indiana University, Bloomington, IN 47408. E-mail: {predrag, fil}@indiana.edu.*
- *Y. Chang is with Jilin University, 828 Xinmin St, GuiLin Lu, Chaoyang Qu, Changchun Shi, Jilin Sheng, 130021, People's Republic of China. E-mail: yichang@ieee.org.*

Manuscript received 7 Sept. 2016; revised 27 Oct. 2017; accepted 23 Dec. 2017. Date of publication 8 Jan. 2018; date of current version 1 June 2018.
(Corresponding author: Makoto Yamada.)
Recommended for acceptance by V.S.S. Tseng.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2018.2789451

to be preserved, while eliminating all others. There are two reasons why the predictive capability of the data may be preserved or even improved when many features are excluded. First, measurements generate many features automatically and their quality is hard to control [6]. Second, biological features are often highly redundant, so that the number of useful features is small. For example, among millions of Single Nucleotide Polymorphisms (SNPs), only a few are useful to predict a certain disease [5]. Although state-of-the-art feature selection algorithms, such as *minimum redundancy maximum relevance* (mRMR) [7], have been proven to be effective and efficient in preparing data for many tasks, they cannot scale up to biological data with millions of features. Moreover, mRMR uses greedy search strategies such as forward selection/backward elimination and tends to produce locally optimal feature sets.

Here we propose a novel feature selection framework for big biological data that makes it possible for the first time to identify very few relevant, non-redundant features among millions. The proposed method is based on two components: *Least Angle Regression* (LARS), an efficient feature selection method [8], and the Hilbert-Schmidt Independence Criterion (HSIC), which enables the selection of features that are non-linearly related [9]. These properties are combined to obtain a method that can exploit *nonlinear* feature dependencies efficiently, and furthermore enables *distributed* implementation on commodity cloud computing platforms. We name our algorithm *Least Angle Nonlinear Distributed* (LAND) feature selection. Experiments demonstrate that the proposed method can reduce the number of features in real-world biological datasets from one million to tens or hundreds, while preserving or increasing prediction accuracy for biological applications.

The following sections present the proposed LAND method in detail and show performance evaluation of LAND on three large, high-dimensional datasets related to the problems of discovering mutations in the tumor suppressor protein p53, classifying cancer phenotypes in a cohort of human prostate cancer patients, and detecting enzymes among protein structures. While existing feature selection methods cannot consider nonlinear dependencies among features in these problems, we show that our approach can reduce the dimensionality by five orders of magnitude.

These results are achieved in minutes to hours of cluster computing time. The selected features are relevant and non-redundant, making it possible to obtain accurate and interpretable models that can be run on a laptop computer.

Contribution:

- We scale up the novel Hilbert-Schmidt Independence Criterion Lasso (HSIC Lasso) [10] to handle ultra high-dimensional and large-scale datasets. To the best of our knowledge, this is the first *minimum redundancy maximum relevance* feature selection method that can handle tens of thousand data samples with millions of features.
- We propose the first implementation of nonlinear feature selection on distributed computing platforms.
- We demonstrate that LAND feature selection outperforms state-of-the-art methods on three real-world, large, high-dimensional datasets.

2 Related Work

In this section, we review existing nonlinear feature selection methods and show their drawbacks.

Maximum Relevance (MR) feature selection is a popular approach that selects m features with the largest relevance to the output [7]. The feature screening method [11] is also an MR-method. Usually, mutual information and kernel-based independence measures such as HSIC are used as relevance score [9]. MR-based methods are simple yet efficient and can be easily applicable to high-dimensional and large sample problems. However, since MR-based approaches only use input-output relevance and not use input-input relevance, they tend to select redundant features, so that the selected features can be very similar to each other. This may not help in improving overall classification/regression accuracy and interpretability.

Minimum Redundancy Maximum Relevance [7] was proposed to deal with the feature redundancy problem; it selects features that have high relevance with respect to an output and are non-redundant. It has been experimentally shown that mRMR outperforms MR feature selection methods [7]. Moreover, there exists an off-the-shelf C++ implementation of mRMR, and it can be applicable to a large and high dimensional feature selection. Fast Correlation Based Filter (FCBF) can also be regarded as an mRMR method, which uses symmetric uncertainty to calculate dependencies of features and finds the best subset via backward selection with sequential search [12]. It has also been reported that FCBF compares favorably with mRMR [13]. However, both mRMR and FCBF use greedy search strategies, such as forward selection/backward elimination, which tend to produce locally optimal feature sets.

Convex relaxed versions of mRMR called Quadratic Programming Feature Selection (QPFS) [14] and spectral relaxation (SPEC) [15] were proposed to obtain a globally optimal feature set. An advantage of QPFS and SPEC over mRMR is that they can find a globally optimal solution by just solving a QP problem. The authors showed that QPFS compares favorably with mRMR for large sample size but low-dimensional cases (e.g., $d < 10^3$ and $n > 10^4$). However, QPFS and SPEC tend to be computationally expensive for large and high-dimensional cases, since they need to compute $d(d-1)/2$ mutual information scores. A Nyström approximation approach was proposed to deal with the computational problem in QPFS [14]. It has been shown experimentally that QPFS with Nyström approximation compares favorably with mRMR both in accuracy and time. However, for large and high-dimensional problems, the computational cost for mutual information is still very high.

Feature selection based on forward/backward elimination with HSIC (FOHSIC/BAHSIC) is also a widely used feature selection method [16]. An advantage of HSIC-based feature selection over mRMR is that the HSIC score can be accurately estimated. Moreover, HSIC can be implemented very easily. However, similar to mRMR, features are selected using a greedy search algorithm leading to locally optimal feature sets. HSFS [17] is a continuously relaxed version of FOHSIC/BAHSIC, designed to obtain a better feature set that could be solved by limited-memory BFGS (L-BFGS) [18]. However, HSFS is a non-convex method; restarting from many different initial points would be necessary to select good features, which is computationally expensive.

For small and high-dimensional feature selection problems (e.g., $n < 100$ and $d > 10^4$), ℓ_1 regularized approaches such as Lasso are useful [19], [20]. In addition, Lasso is known to scale well with both number of samples and dimensionality [19], [20]. However, Lasso can only capture linear dependencies between input features and output values. HSIC Lasso was proposed recently to handle non-linearity [10]. In HSIC Lasso, with specific choices of kernel functions, non-redundant features with strong statistical dependence on the output values can be found in terms of HSIC by simply solving a Lasso problem. Although empirical evidence shows that HSIC Lasso outperforms most existing feature selection methods [10], in general HSIC Lasso tends to be expensive compared to simple Lasso when the number of samples increases. Moreover, the statistical properties of HSIC Lasso are not well studied. Recently, a few *wrapper* feature selection methods were proposed, including the feature generating machine [21] and an SVM based approach [22]. These methods are state-of-the-art feature selection methods for high-dimensional and/or large-scale data. However, wrapper methods are computationally expensive for *ultra* high-dimensional and large-scale datasets.

Sparse Additive Models (SpAM) are useful for high-dimensional feature selection problems [23], [24], [25], [26] and can be efficiently solved by a *backfitting* algorithm [23], resulting in globally optimal solutions. Also, statistical properties of the SpAM estimator are well studied [23]. A potential weakness of SpAM is that it can only deal with additive models and may not work well for non-additive models. Hierarchical Multiple Kernel Learning (HMKL) [27], [28] is a nonlinear feature selection method that can fit complex functions such as non-additive functions. However, the computation cost of HMKL is rather expensive. In particular, since HMKL searches among $(m+1)^d$ combinations of kernels (m is the total number of selected kernels), the computation cost heavily depends on the dimensionality d.

3 Least Angle Nonlinear Distributed Feature Selection

We first formulate the supervised feature selection problem and propose the *Least Angle Nonlinear Distributed* (LAND) feature selection.

3.1 Problem Formulation

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d]^\top \in \mathbb{R}^{d\times n}$ denote the input data, a matrix where a column $\boldsymbol{x}_i \in \mathbb{R}^d$ represents an observation vector composed of d elements (features) and a row $\boldsymbol{u}_j \in \mathbb{R}^n$ represents a feature vector composed of n elements (observations). Let $\boldsymbol{y} = [y_1, \ldots, y_n]^\top \in \mathbb{R}^n$ denotes the output data or labels so that $y_i \in \mathcal{Y}$ is the label for $\boldsymbol{x}_i$. The output domain $\mathcal{Y}$ can be either continuous (as in regression problems) or categorical (as in classification problems).

The goal of supervised feature selection is to find m features ($m \ll d$) that are most relevant for predicting the output $\boldsymbol{y}$ for observations $\boldsymbol{X}$.

To efficiently solve a large and high-dimensional feature selection problem, next we propose a *nonlinear* extension of LARS [8] leveraging HSIC [9]. Then, we introduce an approximation to reduce the memory and computational requirements of the algorithm. This approximation enables our feature selection method to be deployed on a distributed computing platform, scaling up to big biological data.

3.2 HSIC Lasso with Least Angle Regression (LAND)

Let us define the kernel (similarity) matrix of the kth feature observations

$$[\boldsymbol{K}^{(k)}]_{ij} = K(u_{ki}, u_{kj}), \quad i, j = 1, \ldots, n,$$

and outputs

$$[\boldsymbol{L}]_{ij} = L(y_i, y_j), \quad i, j = 1, \ldots, n,$$

where u_{ki} is the ith element of kth feature vector $\boldsymbol{u}_k$ and $K(u, u')$ and $L(y, y')$ are kernel functions. In principle, any universal kernel function such as the Gaussian or Laplacian kernels can be used [9]. Here, we first normalize feature $\boldsymbol{u}$ to have unit standard deviation and then use the Gaussian kernel

$$K(u, u') = \exp\left(-\frac{(u-u')^2}{2\sigma_{\mathrm{u}}^2}\right),$$

where σ_{u} is the kernel width.

For the outputs, in regression cases ($y \in \mathbb{R}$) we similarly normalize y to have unit standard deviation and then use the Gaussian kernel

$$L(y, y') = \exp\left(-\frac{(y-y')^2}{2\sigma_{\mathrm{y}}^2}\right).$$

In this paper, we use $\sigma_{\mathrm{u}} = 1$ and $\sigma_{\mathrm{y}} = 1$. In classification cases (i.e., y is categorical) we use the delta kernel, which has been shown to be useful for multi-class problems [16]:

$$L(y, y') = \begin{cases} 1/n_y & \text{if } \ y = y' \\ 0 & \text{otherwise,} \end{cases}$$

where n_y is the number of observations in class y.

HSIC Lasso [10] is formulated as an optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^d} \ & \left\| \widetilde{\boldsymbol{L}} - \sum_{k=1}^{d} \alpha_k \widetilde{\boldsymbol{K}}^{(k)} \right\|_{\mathrm{F}}^2 + \lambda \|\boldsymbol{\alpha}\|_1, \\ \text{s.t.} \ & \alpha_1, \ldots, \alpha_d \geq 0, \end{aligned} \tag{1}$$

where $\lambda \geq 0$ is a regularization parameter, $\|\cdot\|_1$ is the ℓ_1 norm, $\|\cdot\|_F$ is the Frobenius norm ($\|\boldsymbol{M}\|_{\mathrm{F}} = \sqrt{\mathrm{tr}(\boldsymbol{M}\boldsymbol{M}^\top)}$), and $\widetilde{\boldsymbol{K}}$ and $\widetilde{\boldsymbol{L}}$ are the normalized kernel matrices such that $\boldsymbol{1}_n^\top \widetilde{\boldsymbol{K}} \boldsymbol{1}_n = \boldsymbol{1}_n^\top \widetilde{\boldsymbol{L}} \boldsymbol{1}_n = 0$ and $\|\widetilde{\boldsymbol{K}}\|_{\mathrm{F}}^2 = \|\widetilde{\boldsymbol{L}}\|_{\mathrm{F}}^2 = 1$. In this paper, we employ least angle regression [8] to solve Eq. (1) (See Algorithm 1), and we name this LARS variant *Least Angle Nonlinear Distributed* (LAND) feature selection. Note that, LAND is a LARS variant of the novel Hilbert-Schmidt Independence Criterion Lasso (HSIC Lasso) [10], [29]. The difference between HSIC Lasso and LAND is the optimization technique. In LAND, since we select features one by one, we find a local optimal set. However, in practice, we found that feature selection performance of LAND and HSIC Lasso is basically equivalent (See experimental section).

The solution of the LAND problem enables the selection of the most relevant, least redundant features. To illustrate why, we can rewrite the objective function in Eq. (1) as:

$$1 - 2\sum_{k=1}^{d} \alpha_k \mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{y}) + \sum_{k,l=1}^{d} \alpha_k \alpha_l \mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{u}_l), \qquad (2)$$

where $\mathrm{NHSIC}(\boldsymbol{u}, \boldsymbol{y}) = \mathrm{tr}(\widetilde{\boldsymbol{K}}\widetilde{\boldsymbol{L}})$ is the normalized version of HSIC [30], an independence measure such that $\mathrm{NHSIC}(\boldsymbol{u}, \boldsymbol{y}) = 1$ if $\boldsymbol{u} = \boldsymbol{y}$ and $\mathrm{NHSIC}(\boldsymbol{u}, \boldsymbol{y}) = 0$ if and only if the two random variables $\boldsymbol{u}$ and $\boldsymbol{y}$ are independent (see proof in Section 3.5). The un-normalized HSIC was employed in the original paper [10]. However, HSIC takes some positive value when input and output variables are dependent. That is, $\mathrm{HSIC}(\boldsymbol{u}_k, \boldsymbol{y}) > \mathrm{HSIC}(\boldsymbol{u}_{k'}, \boldsymbol{y})$ may not mean that $\boldsymbol{u}_k$ is more strongly associated with $\boldsymbol{y}$ than $\boldsymbol{u}_{k'}$. Therefore, it is natural to normalize HSIC for feature selection problems.

If output $\boldsymbol{y}$ has high dependence on the kth feature $\boldsymbol{u}_k$, $\mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{y})$ is large and thus α_k should also be large, meaning that the feature should be selected. On the other hand, if $\boldsymbol{u}_k$ and $\boldsymbol{y}$ are independent, $\mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{y})$ is close to zero; α_k should thus be small and the kth feature will not be selected. Furthermore, if $\boldsymbol{u}_k$ and $\boldsymbol{u}_l$ are strongly dependent on each other, $\mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{u}_l)$ is large and thus either α_k or α_l will be small; only one of the redundant features will be selected.

In practice, LAND iteratively selects non-redundant features with a strong relevance for determining the output. To select the kth feature we first need to consider its relevance with respect to the output, indicated by $\mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{y})$. Second, a feature is discounted based on its redundancy with respect to previously selected features, given by $\sum_{i:\alpha_i>0} \alpha_i \mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{u}_i)$. Hence we define the *selection score* of the kth feature as $c_k = \mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{y}) - \sum_{i:\alpha_i>0} \alpha_i \mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{u}_i)$. After the feature is selected, we update the α coefficients.

A key challenge of solving problem (1) is that it requires huge memory ($O(dn^2)$) to store all kernel matrices $\widetilde{\boldsymbol{K}}^{(k)}$. For example in the enzyme dataset described below ($d =$ 1,062,420, $n =$15,328), the naive implementation requires more than a petabyte of memory, which is not feasible. Moreover, the computing time for matrix multiplications scales as $O(mdn^3)$, making it unfeasible when both d and n are large. We address these issues by applying a kernel approximation.

3.3 Nyström Approximation for NHSIC

The Nyström approximation [31] allows us to rewrite $\mathrm{NHSIC}(\boldsymbol{u}, \boldsymbol{y}) = \mathrm{tr}(\widetilde{\boldsymbol{K}}\widetilde{\boldsymbol{L}}) \approx \mathrm{tr}(\boldsymbol{F}\boldsymbol{F}^\top \boldsymbol{G}\boldsymbol{G}^\top)$, where, in the regression case,

$$\boldsymbol{F} = \boldsymbol{\Gamma} \boldsymbol{K}_{nb} \boldsymbol{K}_{bb}^{-1/2} / (\mathrm{tr}((\boldsymbol{K}_{bb}^{-1/2} \boldsymbol{K}_{nb}^\top \boldsymbol{K}_{nb} \boldsymbol{K}_{bb}^{-1/2})^2))^{1/4},$$
$$\boldsymbol{G} = \boldsymbol{\Gamma} \boldsymbol{L}_{nb} \boldsymbol{L}_{bb}^{-1/2} / (\mathrm{tr}((\boldsymbol{L}_{bb}^{-1/2} \boldsymbol{L}_{nb}^\top \boldsymbol{L}_{nb} \boldsymbol{L}_{bb}^{-1/2})^2))^{1/4}.$$

Here, $\boldsymbol{F}\boldsymbol{F}^\top$ is a low-rank approximation of $\widetilde{\boldsymbol{K}}$, such that $\boldsymbol{K}_{nb} \in \mathbb{R}^{n\times b}$ and $[\boldsymbol{K}_{nb}]_{ij} = K(u_i, u_{b,j})$, where $\boldsymbol{u}_b \in \mathbb{R}^b$ is a basis vector (see the experimental section for more details). Analogously, $\boldsymbol{K}_{bb} \in \mathbb{R}^{b\times b}$, $\boldsymbol{L}_{nb} \in \mathbb{R}^{n\times b}$, and $\boldsymbol{L}_{bb} \in \mathbb{R}^{b\times b}$. The parameter b is an upper bound on the rank of the $\boldsymbol{K}_{nb}$ and $\boldsymbol{L}_{nb}$ matrices. The higher b, the better the approximation, but the higher the computational and memory costs. If the number of observations n is very large, we can make the problem tractable by using $b \ll n$ without sacrificing the predictive power of the selected features, as shown the next section. The resulting complexity of kernel computation and multiplication for selecting m features is $O(dbn + mdb^2n) = O(mdb^2n)$. Moreover, for each dimension, we only need to store the $\boldsymbol{F} \in \mathbb{R}^{b\times n}$ matrix, yielding space complexity $O(dbn)$. The approximation reduces the overall time complexity of the algorithm by a factor $O(n^2/b^2)$ and the memory requirements by a factor $O(n/b)$ (see Table 1). Note that the original HSIC Lasso formulation in Eq. (1) can find a globally optimal solution without the Nyström approximation. However, in practice, since each kernel Gram matrix (i.e., $\boldsymbol{K}^{(k)}$) is computed from only one feature, we can accurately approximate each kernel Gram matrix by the Nyström approximation. Thus, we can empirically find a good solution if we set the number of bases in the Nyström approximation relatively large (in this paper, we found b = 10, 20 works well).

TABLE 1
Summary of Computational Complexity and Memory Size of Different Implementations of LAND

Method	Kernel	Multiplication	Memory
Naïve	$O(dn^2)$	$O(mdn^3)$	$O(dn^2)$
Nyström	$O(dbn)$	$O(mb^2dn)$	$O(dbn)$
Map-Reduce	$O(dbn/M)$	$O((mb^2/M)dn)$	$O(dbn)$

The total time complexity is obtained by adding the kernel computation and multiplication times, which are dominated by the latter.

In the classification case, we can use the above technique to approximate the kernel matrix $\boldsymbol{F}$ and compute $\boldsymbol{G}$ as

$$\boldsymbol{G}_{k,j} = \begin{cases} \frac{1}{\sqrt{n_k}} & \text{if } \ k = y_j \\ 0 & \text{otherwise,} \end{cases}$$

where $\boldsymbol{G} \in \mathbb{R}^{C\times n}$ and C is the number of classes. The computational complexity of kernel computation and multiplication is $O(mbdn(b + C))$ and the memory complexity is $O(dn(b + C))$. These too are dramatic reductions in computational time and memory.

3.4 Distributed Computation

While the Nyström approximation is useful for data with many observations (large n), the computational cost of LAND makes it unfeasible on a single computer for ultra high-dimensional cases, i.e., when the number of features is extremely large (e.g., $d \geq 10^6$). Fortunately, we can compute the kernel matrices $\{\boldsymbol{F}_k\}_{k=1}^d$ in parallel. The selection scores c_k can be computed independently as well. These properties make it possible to further speed up LAND with a distributed computing framework. The resulting computational complexity is $O(mdb^2n/M)$, where M is the number of mappers (Table 1).

The proposed LAND algorithm is implemented on a cluster for scalability to large datasets. Map-Reduce is a widely adopted distributed computing framework. It consists of a *map* procedure that breaks up the problem into many small tasks that can be performed in parallel, and distributes these tasks to multiple computing nodes (*mappers*). A *reduce* procedure then is executed on multiple nodes (*reducers*) to aggregate the computed results. For example, we denote the map function with inputs $\{\boldsymbol{F}_k\}_{k=1}^d, \boldsymbol{G}$ and the corresponding reduce function as

$$\mathrm{map}(\{\boldsymbol{F}_k\}_{k=1}^d, \boldsymbol{G} : \langle k, \mathrm{tr}((\boldsymbol{F}_k^\top \boldsymbol{G})^2)\rangle)$$
$$\mathrm{reduce}(\langle k,\, \mathrm{tr}((\boldsymbol{F}_k^\top \boldsymbol{G})^2)\rangle :\, f_k = \mathrm{tr}((\boldsymbol{F}_k^\top \boldsymbol{G})^2))$$

where the map function returns key-value pairs and the reduce function stores the key-value pairs into a vector f_k.

We employ two Map-Reduce frameworks. Hadoop (hadoop. apache.org) is used for computing $\boldsymbol{F}_k$'s in the Nyström approximation. Spark (spark.apache.org) reduces the data access cost by storing intermediate results in memory, and is used for the iteration operations. Our Map-Reduce implementation is shown in Algorithm 1.

Algorithm 1. LAND (Map-Reduce Spark Version)

```
Initialize: $\boldsymbol{\alpha} = \mathbf{0}_\mathrm{d}$, $\mathcal{A} = []$ (active set), and $\mathcal{I} = \{1, 2, \ldots, d\}$
(non-active set).
Compute $\boldsymbol{G}$ and store it in memory.
/* Compute $\{\boldsymbol{F}_k\}_{k=1}^d$ and store them in memory. */
$\{\boldsymbol{F}_k\}_{k=1}^d = \mathrm{map}(\{\boldsymbol{u}_k\}_{k=1}^d : <k, \boldsymbol{F}_k>)$

/* Compute $\mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{y})$ */
$\mathrm{map}(\{\boldsymbol{F}_k\}_{k=1}^d, \boldsymbol{G} : <k, \mathrm{tr}(\boldsymbol{F}_k^\top \boldsymbol{G})^2>)$
$\mathrm{reduce}(<k, \mathrm{tr}((\boldsymbol{F}_k^\top \boldsymbol{G})^2)>: f_k = \mathrm{tr}((\boldsymbol{F}_k^\top \boldsymbol{G})^2))$

/* NHSIC coefficient matrix */
$\boldsymbol{R} = []$
/* Select $m$ features */
while $|\mathcal{A}| < m$ do
   /* Compute $c_k = \mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{y}) - \sum_{i=1}^d \alpha_i \mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{u}_i)$ */
   $\boldsymbol{c} = \boldsymbol{f} - \boldsymbol{R}\boldsymbol{\alpha}_\mathcal{A}$

   Find feature index: $j = \mathrm{argmax}_{\boldsymbol{c}_\mathcal{I}}\, c_k > 0$
   Update sets: $\mathcal{A} = [\mathcal{A}\ \ j], \mathcal{I} = \mathcal{I} \backslash j$
   Update coefficients:

      $\boldsymbol{\alpha}_\mathcal{A} = \boldsymbol{\alpha}_\mathcal{A} + \widehat{\mu} \boldsymbol{Q}_\mathcal{A}^{-1} \mathbf{1},$
      $[\boldsymbol{Q}_\mathcal{A}]_{i,j} = \mathrm{NHSIC}(\boldsymbol{u}_{\mathcal{A},i}, \boldsymbol{u}_{\mathcal{A},j})$
      $\widehat{\mu} = \min_{\mu} \begin{cases} \exists \ell \in \mathcal{I} : \widetilde{c}_\ell = \boldsymbol{c}_\mathcal{A} \\ \boldsymbol{c}_\mathcal{A} = \mathbf{0}, \end{cases}$

   /* Compute $\{\mathrm{NHSIC}(\boldsymbol{u}_j, \boldsymbol{u}_k)\}_{k=1}^d$ */
   $\mathrm{map}(\{\boldsymbol{F}_k\}_{k=1}^d, \boldsymbol{F}_j : <k, \mathrm{tr}(\boldsymbol{F}_k^\top \boldsymbol{F}_j)^2>)$
   $\mathrm{reduce}(<k, \mathrm{tr}(\boldsymbol{F}_k^\top \boldsymbol{F}_j)^2>: r_{k,j} = \mathrm{tr}(\boldsymbol{F}_k^\top \boldsymbol{F}_j)^2>)$
   $\boldsymbol{R} = [\boldsymbol{R}\ \ \boldsymbol{r}_j]$
end while
```

In Section 4.4, we use $b = 20$ for the p53 data and the prostate cancer data, and $b = 10$ for the enzyme data.

3.5 Relation to High-Dimensional Feature Screening Method

Let us establish a relation between the proposed method, LAND, and the feature screening method [2]. Feature screening is a maximum relevance [15] approach used widely in the statistics community. It aims to select a subset of features with the goal of dimensionality reduction, without affecting the statistical properties of the data. The idea is to rank the covariates between the input variables $\boldsymbol{u}$ and the output response $\boldsymbol{y}$ according to some degree of dependence. For example, one can choose NHSIC as an independence measure and rank the d features $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d$ according to the values of $\mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{y})$. The top m features are then selected. This MR-NHSIC baseline can be regarded as a feature screening method.

Proposition 1. *If any pair of features $\boldsymbol{u}_k$ and $\boldsymbol{u}_{k'}$ are assumed to be independent, then there exists a pair (λ, m) such that the top m features obtained by the feature screening method [2] are the same of those obtained by solving Eq. (1).*

Proof. According to Theorem 4 by Gretton et al. [14], $\mathrm{HSIC}(\boldsymbol{u}_k, \boldsymbol{u}_{k'}) = 0$ if and only if two features $\boldsymbol{u}_k$ and $\boldsymbol{u}_{k'}$ are independent. Hence, if the pair of features $\boldsymbol{u}_k$ and $\boldsymbol{u}_{k'}$ is independent, we have the following result using the definition of NHSIC:

$$\begin{aligned} \mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{u}_{k'}) &= \mathrm{tr}(\bar{\boldsymbol{K}}^{(k)} \bar{\boldsymbol{K}}^{(k')}) \\ &= \frac{\mathrm{HSIC}(\boldsymbol{u}_k, \boldsymbol{u}_{k'})}{\sqrt{\mathrm{tr}(\bar{\boldsymbol{K}}^{(k)} \bar{\boldsymbol{K}}^{(k)})}\sqrt{\mathrm{tr}(\bar{\boldsymbol{K}}^{(k')} \bar{\boldsymbol{K}}^{(k')})}} \\ &= \begin{cases} 0 & \text{if } k \neq k' \\ 1 & \text{if } k = k', \end{cases} \end{aligned} \tag{3}$$

where $\mathrm{HSIC}(\boldsymbol{u}_k, \boldsymbol{u}_{k'}) = \mathrm{tr}(\bar{\boldsymbol{K}}^{(k)} \bar{\boldsymbol{K}}^{(k')})$. Note that, $\mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{u}_k) = 1$. Since the two features $\boldsymbol{u}_k$ and $\boldsymbol{u}_{k'}$ are assumed to be independent (i.e., $\| \sum_{k=1}^d \alpha_k \widetilde{\boldsymbol{K}}^{(k)} \|_\mathrm{F}^2 = \|\boldsymbol{\alpha}\|_2^2$) and $\|\bar{\boldsymbol{L}}\|_\mathrm{F}^2 = \mathrm{NHSIC}(\boldsymbol{y}, \boldsymbol{y}) = 1$ by the definition of NHSIC (Eq. (3)), the optimization problem in Eq. (1) is equivalent to:

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathbb{R}^d} \ & \sum_{k=1}^d \alpha_k \mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{y}) - \frac{1}{2}\|\boldsymbol{\alpha}\|_2^2 - \frac{\lambda}{2}\|\boldsymbol{\alpha}\|_1, \\ \text{s.t.} \ & \alpha_1, \ldots, \alpha_d \geq 0. \end{aligned} \tag{4}$$

Next we prove by contradiction that the largest NHSIC values correspond to the largest α_k values in the solution of Eq. (4). Suppose there exists a pair (i, j) such that $\mathrm{NHSIC}(\boldsymbol{u}_i, \boldsymbol{y}) > \mathrm{NHSIC}(\boldsymbol{u}_j, \boldsymbol{y})$ and $\alpha_i < \alpha_j$. Then one can simply switch the values of α_i and α_j to obtain a higher value in the objective function of Eq. (4). This contradiction proves that the largest α_k correspond to the largest values of $\mathrm{NHSIC}(\boldsymbol{u}_k, \boldsymbol{y})$. □

The above proposition draws the connection to high-dimensional feature screening [11]. Since the feature screening method tends to select redundant features, an iterative screening approach is used to filter out redundant features. Fig. 4 shows the obtained (λ, m) pairs of the ASU datasets by LAND.

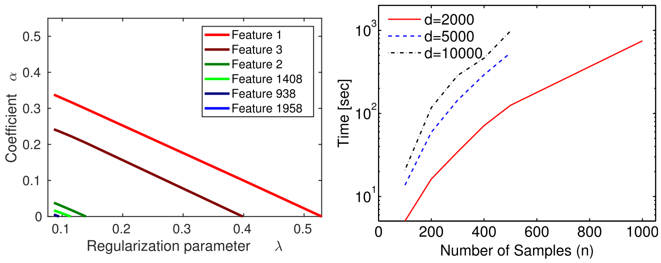
4 Experiments

We first illustrate LAND on synthetic data and small-scale benchmark datasets. Then, we evaluate LAND using three biological datasets with d ranging from thousands to over a million features.

4.1 Evaluation Metrics

We employ the average area under the ROC curve (*AUC*) as a measure of accuracy that is robust with respect to unbalanced classes [32]. Values above 0.5 indicate better-than-random performance; one signifies perfect accuracy.

Let us also define the *dimensionality reduction* as $1 - \frac{m}{d}$ where zero represents the original full set of features and higher values indicate smaller sets of selected features.



(A) Regularization path (B) Computational time

Fig. 1. Performance of LAND on synthetic data. (A) The regularization path, which describes the transition of parameters over the regularization parameter λ in Eq. (1). (B) Computational time (without using the Nyström approximation and the distributed implementation) versus the number of observations n, for different values of the dimensionality d.

Finally, to check whether an algorithm can successfully select non-redundant features, we define the *independence rate*:

$$I = 1 - \frac{1}{m(m-1)} \sum_{\boldsymbol{u}_k, \boldsymbol{u}_l, k > l} |\rho_{k,l}|,$$

where $\rho_{k,l}$ is the Pearson correlation coefficient between the kth and lth features. A large I means that the selected features are more independent, or less redundant. In fact, I is closely related to the redundancy rate [20]. Note that, we employed the linear independence rate, but not nonlinear one such as mutual information or HSIC, since we want to compare all nonlinear feature selection methods fairly. For example, we may be able to get higher independence rate for proposed method than the others if we use HSIC as an independence measure.

4.2 Synthetic Dataset

We consider a regression problem from a 2000-dimensional input, where input data $(X_1, \ldots, X_{2000})$ includes three groups of variables. The first group of three variables (X_1, X_2, X_3) are relevant for the output Y, which is generated according to the following expression:

$$Y = X_1 * \exp(X_2) + X_3 + 0.1 * E,$$

where $E \sim N(0,1)$. All variables are normally distributed. In particular, for the first 1,000 variables, $(X_1, \ldots, X_{1000})^\top \sim N(\boldsymbol{0}_{1000}, \boldsymbol{I}_{1000})$. Here, $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multi-variate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. We define the remaining 1,000 variables as: $X_{1001} = X_1 + 0.01 * E, \ldots, X_{1003} = X_{1003} + 0.01 * E$. The second group of variables $(X_4, \ldots, X_{1000})$ and $(X_{1004}, \ldots, X_{2000})$ are uncorrelated with the output, and therefore irrelevant. The third group X_{1001}, X_{1002}, and X_{1003} are redundant features of X_1, X_2,, and X_3, respectively.

Fig. 1A shows the regularization path for 10 features, and this illustrates that LAND can select non-redundant features. Note that, the values of feature 1,408, 938, and 1,958 are quite small compared to important features. Fig. 1B plots the computational time for LAND on a Xeon 2.4 GHz (16 cores) with 24 GB memory. As can be seen, the computational cost of LAND without using the Nyström approximation and distributed computing increases dramatically with the number of observations. Moreover, since LAND needs $O(dn^2)$ memory space, it is not possible to solve LAND even if the number of observations is small $(n = 1{,}000)$. Thus, the Nyström approximation and distributed computation are necessary for the proposed method to solve high-dimensional and large sample cases.

TABLE 2
Summary of Benchmark Datasets

Type	Dataset	Features (d)	Samples (n)
Classification	AR10P	2,400	130
	PIE10P	2,400	210
	PIX10P	10,000	100
	ORL10P	10,000	100
	TOX	5,748	171
	CLL-SUB	11,340	111
Regression	TRIM32	31,098	120
	Doxorubicin	13,321	99
	Gemcitabine	13,321	99
	Docetaxel	13,321	99

4.3 Benchmark Datasets

Here, we evaluate the accuracy of LAND using real-world benchmark datasets (see Table 2 for details).

4.3.1 Classification

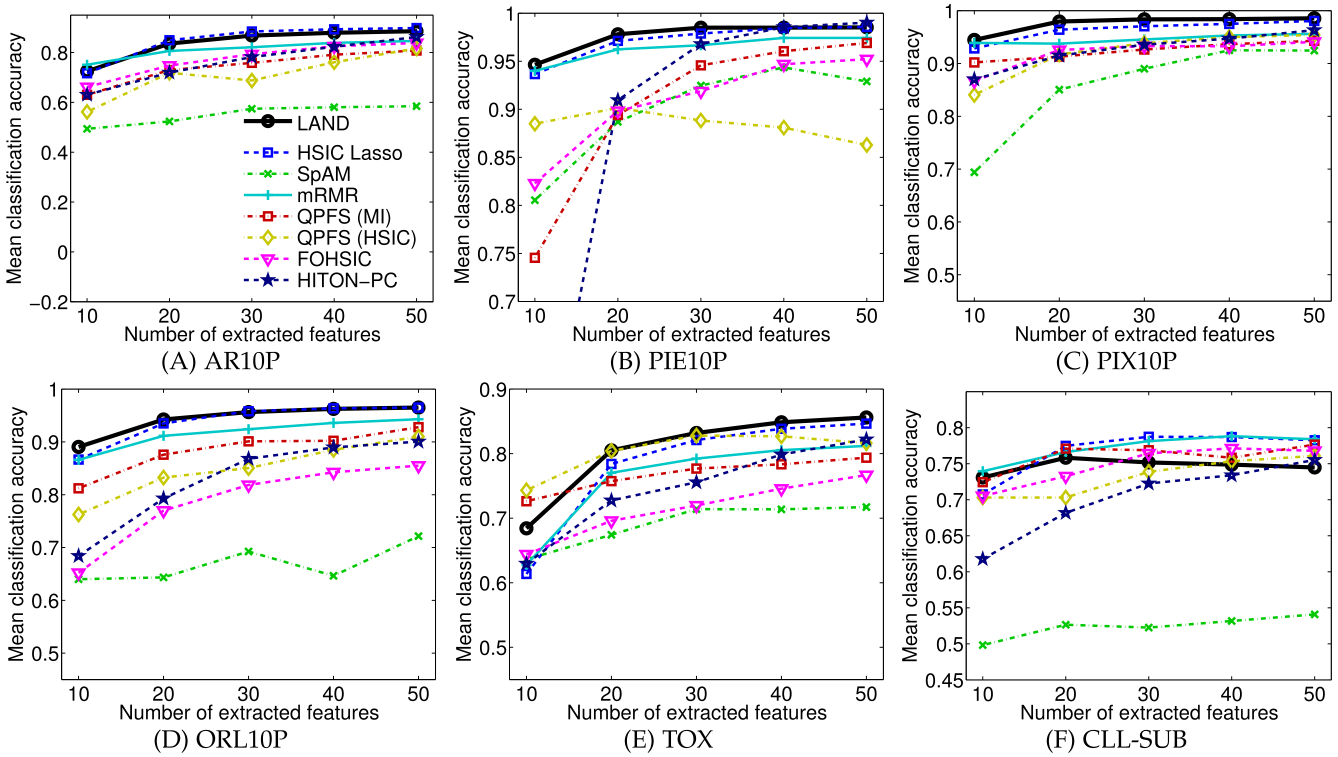
We first consider classification benchmarks[1] with relatively small d and n, allowing us to compare LAND with several baseline methods that are computationally slow. For these classification experiments, we use 80 percent of samples for training and the rest for testing. In each experiment, we apply feature selection on the training data to select the top $m = 10, 20, \ldots, 50$ features and then measure accuracy using the selected features in the test data. We run the classification experiments 100 times by randomly selecting training and test samples and report the average classification accuracy. Since all datasets are multi-class, we use multi-class kernel logistic regression (KLR) [33]. For KLR we use a Gaussian kernel where the kernel width and the regularization parameter are chosen based on 3-fold cross-validation.

Figs. 2A, 2B, 2C, 2D, 2E, and 2F show the average classification accuracy versus the number of selected features. Fig. 3 shows the confusion matrices of the LAND algorithm. With the single exception of the CLL-SUB benchmark, LAND compares favorably with all baselines, including HSIC Lasso, a state-of-the-art high-dimensional feature selection method. For (simple) image classification tasks such as AR10P and PIE10P, we can get high classification accuracy. However, for biology related data (i.e., TOX, CLL-SUB), the classification accuracy tends to be lower than that of the image datasets (PIE10P, PIX10P, ORL10P). This is basically due to the difficulty of biological classification problems. To improve the performance, we may need to learn features from data (e.g., Deep learning). However, learning features from small number of samples are notoriously hard, and an open question in deep learning.

4.3.2 Regression

TRIM32 Data: Our benchmark is the Affymetric GeneChip Rat Genome 230 2.0 Array dataset [34]. In this dataset, there are 120 rat subjects $(n = 120)$. The real-valued expression for

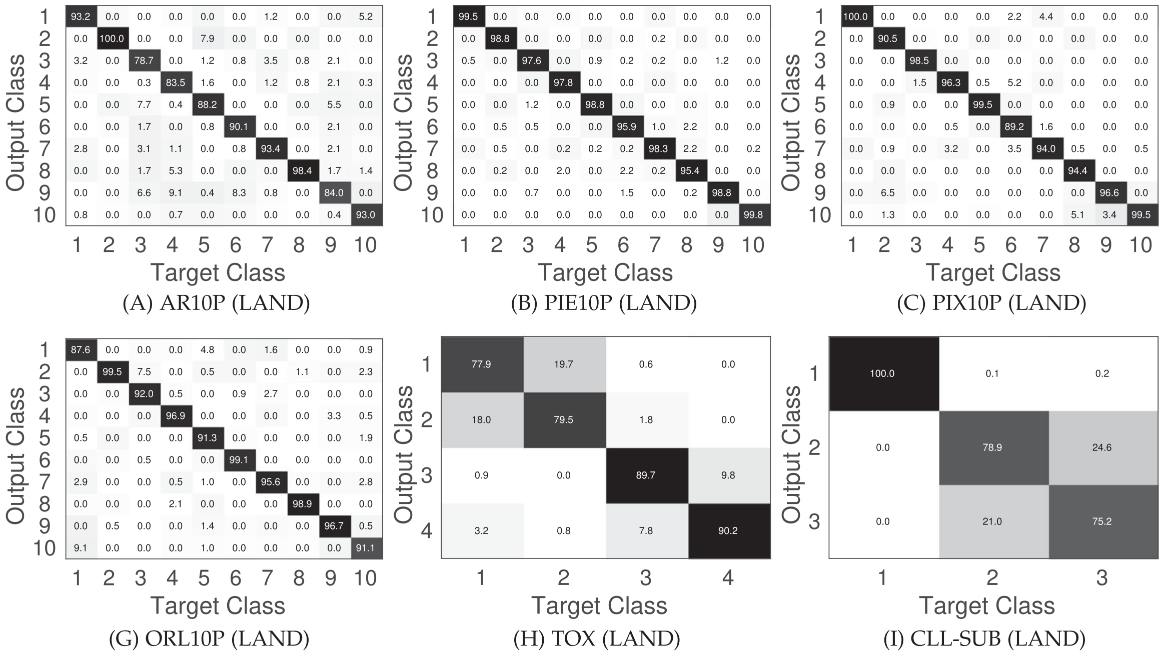
[1] http://featureselection.asu.edu/datasets.php



(A) AR10P (B) PIE10P (C) PIX10P

(D) ORL10P (E) TOX (F) CLL-SUB

Fig. 2. The results for the benchmark datasets. Mean classification accuracy for six classification benchmark datasets. The horizontal axis denotes the number of selected features, and the vertical axis denotes the mean classification accuracy.



(A) AR10P (LAND) (B) PIE10P (LAND) (C) PIX10P (LAND)

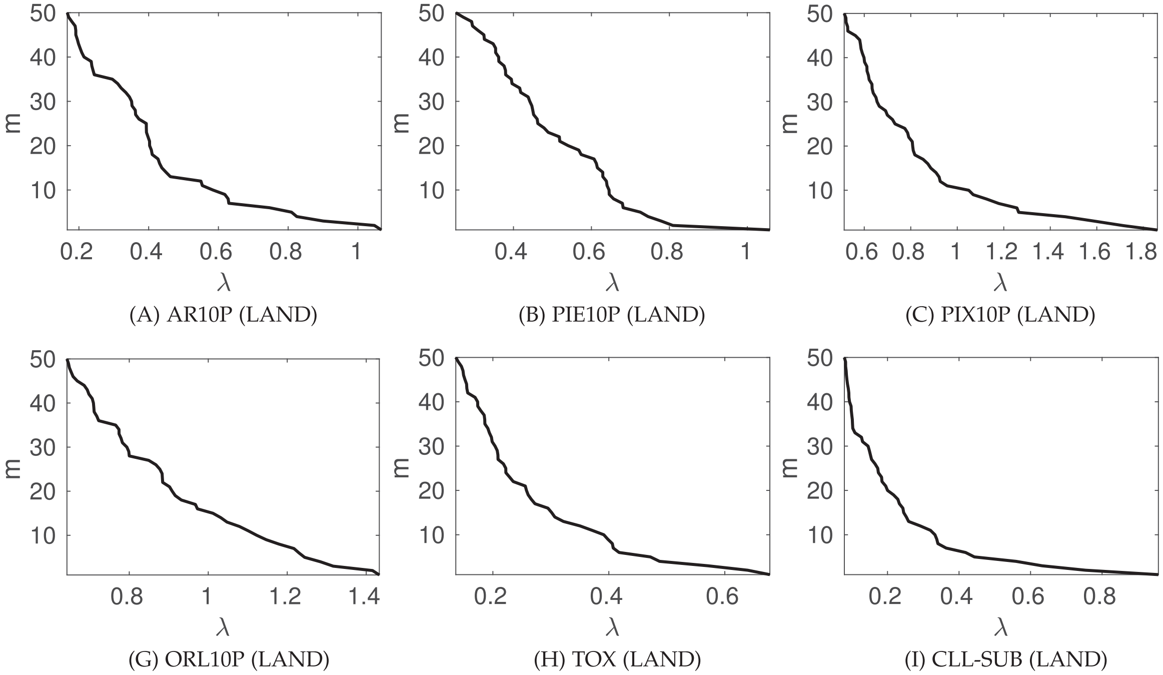
(G) ORL10P (LAND) (H) TOX (LAND) (I) CLL-SUB (LAND)

Fig. 3. The confusion matrices for the benchmark datasets. Mean classification accuracy for six classification benchmark datasets.

over 30 thousand genes ($d = 31{,}098$) in the mammalian eye is important for eye disease. In this paper, we focus on finding genes that are related to the TRIM32 gene [34], [35], which was recently found to cause the Bardet-Biedl syndrome.

For this regression experiment, we use 80 percent of samples for training and the rest for testing. We again select the top $m = 10, 20, \ldots, 50$ features having the largest absolute regression coefficients in the training data. As earlier, we run the regression experiments 100 times by randomly selecting training and test samples, and compute the average mean squared error. We employ kernel regression [31] with the Gaussian kernel. The Gaussian width and the regularization parameter are chosen based on 3-fold cross-validation. In this experiment, most existing methods are too slow to finish. Thus, we only include the LAND, HSIC Lasso, linear Lasso, and mRMR results.

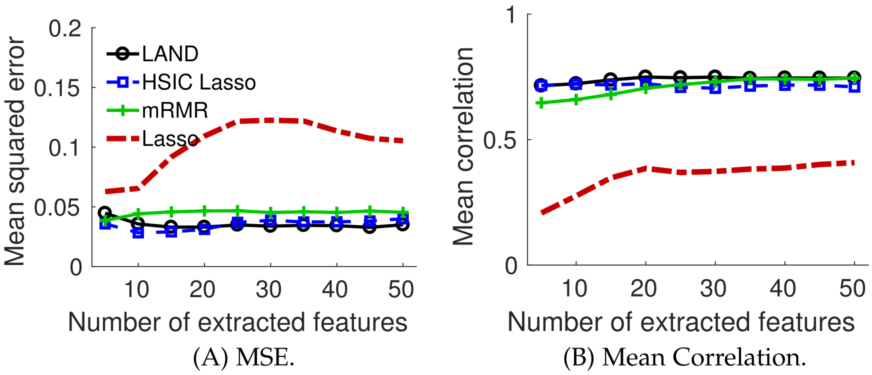
Fig. 5 show the mean squared error and the mean correlation over 100 runs as a function of the number of selected



(A) AR10P (LAND) (B) PIE10P (LAND) (C) PIX10P (LAND)

(G) ORL10P (LAND) (H) TOX (LAND) (I) CLL-SUB (LAND)

Fig. 4. The (λ, m) pair of the benchmark datasets.



(A) MSE. (B) Mean Correlation.

Fig. 5. The results for the TRIM32 dataset. (A) Mean squared error for the TRIM32 data. (B) Mean correlation for the TRIM32 data. The horizontal axis denotes the number of selected features, and the vertical axis denotes the mean squared error (lower is better) and the mean correlation (higher is better).

features. As can be observed, the accuracy obtained with features selected by LAND is better than Lasso, comparable with mRMR and HSIC Lasso. This is because in this regression experiment, the HSIC measure of independence performs better than NHSIC. If we use HSIC instead of NHSIC, LAND can achieve the same accuracy as HSIC Lasso.

Sanger Data Sets: We applied LAND to identify potential driver genes of anti-cancer drug sensitivity in the publicly available "Sanger Genomics of Drug Sensitivity in Cancer dataset form the Cancer Genome Project"[2]. In this dataset, there are 99 anti-cancer drugs ($n = 99$). We considered the real-valued expression for over 10 thousand genes ($d = 13{,}321$) as input and the IC50 values of anti-cancer drugs to reveal driver genes as output. To evaluate the proposed methods, we identified potential driver genes using LAND. We focused on three popular anti-cancer drugs: Doxorubicin, Gemcitabine, and Docetaxel which have attracted considerable attention for cancer research.

Fig. 6 shows the mean squared error and the mean correlation over 100 runs as a function of the number of selected features. Similar to the TRIM32 experiment, LAND outperforms Lasso and compares favorably with HSIC Lasso.

4.4 High Dimensional and Large-Scale Datasets

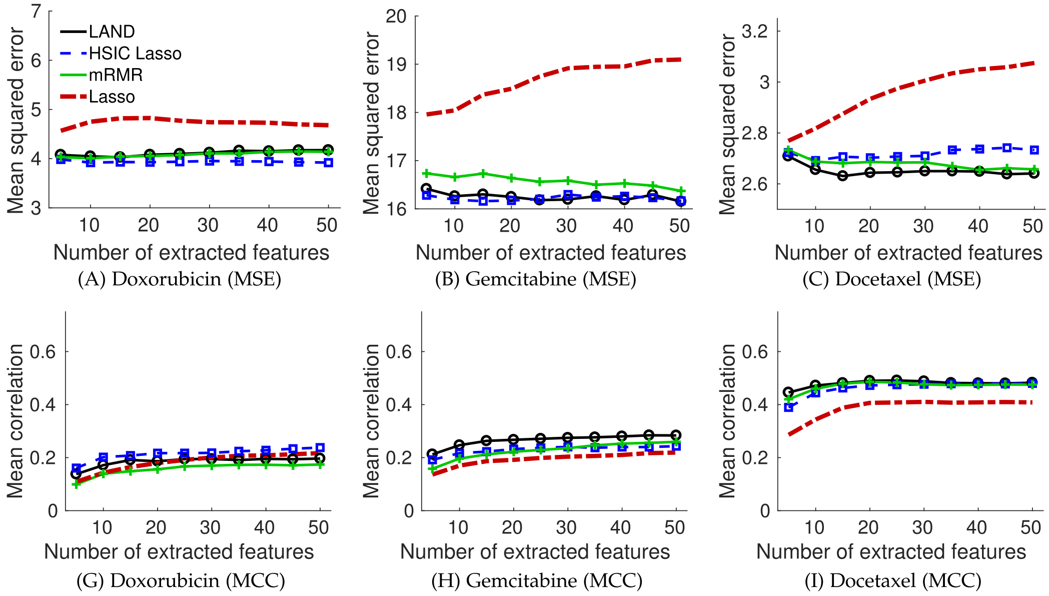
In this section, we evaluate LAND using real-world high-dimensional and/or large-scale datasets. For LAND and MR-NHSIC, we employed the Yahoo research cluster.

We evaluate all feature selection methods by passing the selected features to a supervised learning algorithm. For this purpose we employ gradient boosting decision trees (GBDT) [36] as an off-the-shelf nonlinear classifier.

4.4.1 *Prediction of p53 Transcriptional Activities*

We first consider the p53 mutant dataset [37], where the goal is to predict whether any of $n = 31{,}420$ mutations is active or inactive based on $d = 5{,}408$ features. Class labels are obtained via *in vivo* assays [37]. For this data, we compared LAND with Lasso, mRMR, and MR-NHSIC baselines

[2] http://www.cancerrxgene.org



(A) Doxorubicin (MSE) (B) Gemcitabine (MSE) (C) Docetaxel (MSE)

(G) Doxorubicin (MCC) (H) Gemcitabine (MCC) (I) Docetaxel (MCC)

Fig. 6. The results for the sanger datasets. (A) Mean squared error for the TRIM32 data. (B) Mean correlation for the TRIM32 data. The horizontal axis denotes the number of selected features, and the vertical axis denotes the mean squared error (lower is better) and the mean correlation (higher is better).

on the task of selecting $m = 100$ features. We present results for LAND using $b = 20 \ll n$ and setting the basis vector heuristically to $\boldsymbol{u}_b = [-5, -4.47, \ldots, 4.47, 5.0]^\top \in \mathbb{R}^{20}$.

Accuracy and independence rate metrics for features selected by LAND are compared with those obtained by three state-of-the-art feature selection baselines. We split the data into 26,420 observations used for training the learning algorithm and 5,000 observations for testing. We run the classification experiments 20 times by randomly selecting training and test samples, and report the average accuracy and independence rate metrics. We select the m most relevant features ($m = 10, 20, \ldots, 100$) and employ 100 trees with 20 nodes in the GBDT classifier.

untimes are shown in Table 3. Given the small dimensionality d of this problem, the speed up from distributed feature selection algorithms is not sufficient to offset their cluster overhead. Therefore the single-machine linear algorithm (Lasso) is the most efficient. LAND, however, selects better features. Fig. 7A shows that high accuracy (80 percent AUC) can be achieved by LAND with a dimensionality reduction of over 99 percent. Considering more than $m = 20$ features yields marginal improvements in accuracy. If one selects a small number of features, LAND outperforms the state-of-the-art nonlinear methods in accuracy. Conversely, the same accuracy can be achieved with fewer features. For this small-size benchmark, the performance of the state-of-the-art in linear feature selection (Lasso) is comparable. Fig. 7B plots the independence of the selected features versus m. The features selected by LAND are significantly less redundant compared to the baselines, irrespective of the dimensionality reduction. In summary, LAND selects the most independent features and achieves the top accuracy.

TABLE 3
Sizes of Biological Datasets and Computational Times (in Seconds) to Select 100 Features Using the Nonlinear Methods

Dataset	d	n	MR-NHSIC	mRMR	LAND
p53	5408	26120	290	544	1709
PC	276322	302	383	4018	1284
Enzyme	1062420	13794	5328	n/a	10630

4.4.2 Subnetwork Markers for Prostate Cancer Classification

Next, we applied our approach to a cohort of $n = 383$ prostate cancer (PC) patients. We aim to separate malignant tumors from normal tissues based on $d = 276{,}322$ features (Fig. 8). For LAND, we used $b = 20 \ll n$ and $\boldsymbol{u}_b = [-5, -4.47, \ldots, 4.47, 5.0]^\top \in \mathbb{R}^{20}$.

Accuracy and independence rate of features selected by LAND are compared with those obtained by the three baselines. We split the data into 344 patients used for training the learning algorithm and 39 patients for testing. We ran

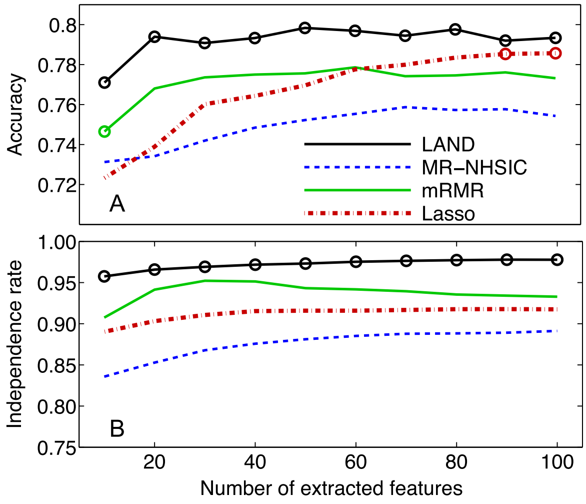


Fig. 7. Results on the p53 benchmark. Circles indicate the methods achieving the best average accuracy/independence rate according to a one-tailed t-test ($p < 0.05$). (A) Accuracy versus number of selected features m. Differences between LAND and all baselines are statistically significant for all dimensionality reduction levels except $m = 10$ (versus mRMR) and $m \geq 90$ (versus Lasso). (B) Independence rate versus m: all differences are significant as indicated by circles.

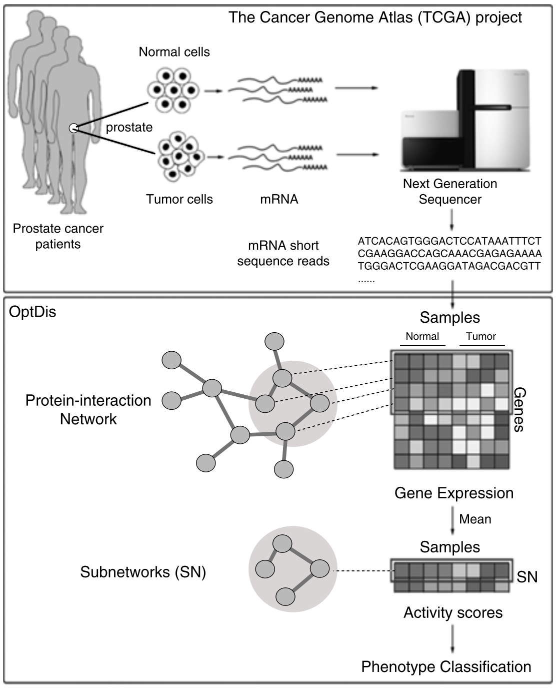

The Cancer Genome Atlas (TCGA) project
Normal cells
prostate
Tumor cells
mRNA
Next Generation Sequencer
Prostate cancer patients
mRNA short sequence reads
ATCACAGTGGGACTCCATAAATTTCT
CGAAGGACCAGCAAACGAGAGAAAA
TGGGACTCGAAGGATAGACGACGTT
......
OptDis
Samples
Normal
Tumor
Genes
Protein-interaction Network
Gene Expression
Mean
Samples
Subnetworks (SN)
SN
Activity scores
Phenotype Classification

Fig. 8. Overview of prostate cancer data. As part of The Cancer Genome Atlas (TCGA) project [38], mRNA sequence based gene expression profiles of tumor and normal cells were obtained from a cohort of prostate adenocarcinoma patients. The data was downloaded from the TCGA data portal. Using OptDis [39], we extracted connected subnetworks of maximum size 7 from the STRING v10 protein-protein interaction network [40] using only edges with score above 0.9. We finally used the average expression of the component genes of each subnetwork as a feature.

the classification experiments 20 times by randomly selecting training and test observations, and report average performance. We select m most relevant features ($m = 10, 20, \ldots, 100$). The classifier employs 100 trees with 20 nodes.

As shown in Fig. 9A, LAND achieves the best accuracy (AUC above 95 percent) with as few as $m = 80$ features—a dimensionality reduction over 99.97 percent. Lasso selects more independent features (Fig. 9B), however its accuracy is lower. As shown in Table 3, LAND is more efficient than mRMR on a cluster. The time required by mRMR increases dramatically with the dimensionality of the data, while the computing time for LAND does not. MR-NHSIC is even faster, however it does not select independent features (Fig. 9B).

4.4.3 Enzyme Protein Structure Detection

Our third task is to distinguish between enzyme and non-enzyme protein structures. Enzyme data contains all homomeric protein structures from the Protein Data Bank (PDB) [41] as of February 2015 such that (i) each structure is at least 50 amino acid residues long; and (ii) protein structure is determined by X-ray crystallography with resolution below 2.5Å. All proteins with 100 percent sequence identity to any other protein in the dataset are filtered out. In the case of multiple exact matches, the structure with best resolution is selected. To generate the features, the protein structures are modeled as contact graphs, where each residue is represented as a labeled vertex and two spatially close residues, with euclidean distance between any two atoms below 4.5Å, are linked by an undirected edge. Each feature is obtained by counting labeled rooted graphlets with up to four vertices [42]. More details on rooted graphlets can be found in the literature [43], [44], [45]. There are $n = 15{,}328$ observation vectors (7,767 enzymes and 7,561 non-enzymes), each of dimensionality $d = 1{,}062{,}420$. We split the observations into 90 percent (13,794) for training and 10 percent (1,534) for testing. We report average performance across five random splits. We use classifiers with 500 trees and 20 nodes, and select $m = 5, 10, 20, \ldots, 100$ features.

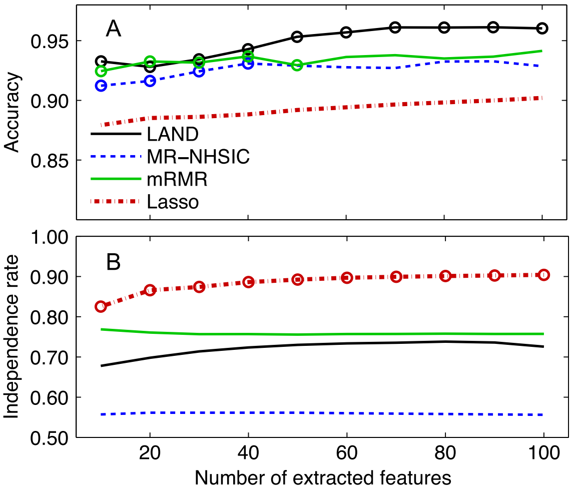


Fig. 9. Results on the prostate cancer dataset. Circles indicate the methods achieving the best average accuracy/independence rate according to a one-tailed t-test ($p < 0.05$). (A) Accuracy versus number of selected features m. Differences between LAND and the baselines are statistically significant for $m \geq 60$. (B) Independence rate versus m: All differences are significant.

To explore the complexity stemming from the ultra-high dimensionality of this problem, we trained a state-of-the-art classifier based on the graph kernel method [42] on the full dataset. The resulting model achieved high accuracy (AUC above 90 percent), but needed roughly 18 days of computing time using our fastest server (a machine with 64 2.4 GHz processors and 512 GB of RAM). This demonstrates the need for feature selection.

The naive version of LAND requires memory that scales as as $O(dn^2)$ (cf. Table 1) for storing the kernel matrices. Due to the very large number of features and the large number of observations in this dataset, this is prohibitive—approximately 1.5 petabytes. By using $b = 10 \ll n$, we reduce the space complexity to $O(dbn)$ (cf. Table 1) and the memory requirement to a more manageable one terabyte for LAND. We also use $\boldsymbol{u}_b = [-5, -3.89, \ldots, 3.89, 5.0]^\top \in \mathbb{R}^{10}$.

Due to the size of this dataset, running the mRMR baseline would require hundreds of gigabytes of memory. Since this is unfeasible, we only compare LAND with one nonlinear baseline (MR-NHSIC). LAND achieves higher accuracy when selecting very few features, reducing the dimensionality of the problem by over 99.99 percent (Fig. 10A). This also implies more interpretable results and an enormous speed up in classification/prediction time. Accuracy is only slightly lower than what is obtained with the state-of-the-art classifier using all d features. Two linear (Lasso) baselines yielded lower accuracy with a worse dimensionality reduction. The features selected by LAND have lower redundancy than those selected by the baseline (Fig. 10B). Although Lasso does not achieve good performance, it is the most efficient and can select features in 973 seconds. Fig. 10C compares the

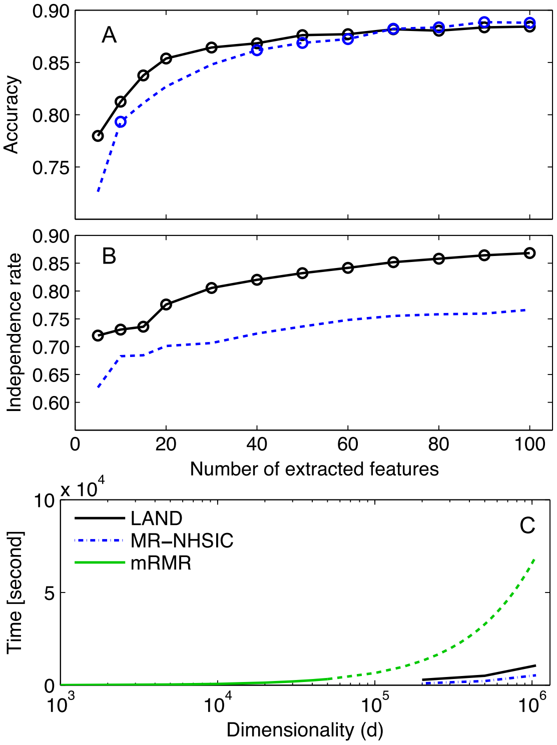


Fig. 10. Results for the enzyme dataset. Circles indicate best accuracy/independence rate according to t-tests ($p < 0.05$). (A) Differences in accuracy are significant for $m = 5, 20, 30, 40$. We also ran two linear Lasso baselines (not shown) with regularization parameter $\lambda = 10^{-6}$ and 0.5×10^{-6}. These settings yield low accuracy (AUC between 0.75 and 0.85) with many features (an average of $m =$ 1,665 and 7,675, respectively). (B) Independence rate versus m: all differences are significant. (C) Feature selection runtime versus d. We ran the algorithms on reduced-dimensionality datasets. We estimated the mRMR runtime for $d >$ 50,000, given that it scales linearly with d (dotted line).

computational time of LAND, MR-NHSIC, and mRMR. LAND runs in just 3 hours on a Yahoo computer cluster—about double the MR-NHSIC runtime (cf. Table 3). This is a trade-off for the independence of the selected features.

5 Discussion

The proposed LAND feature selection method is guaranteed to find an optimal solution, and does so efficiently by exploiting a non-negative variant of the LARS algorithm where the parameter space is sparse. Our experimental results demonstrate that LAND scales up to large and high-dimensional biological data by allowing a distributed implementation on commodity cloud computing platforms.

Let us further investigate the structural motifs identified by LAND from the enzyme data (Section 4.4.1). Our initial requirement was that the method be able to identify catalytic site-related features, as well as other features relevant for the signatures of enzymatic structures. One of the most interesting motifs is the DH graphlet, containing aspartic acid and histidine, which is also the most commonly seen motif in the Catalytic Site Atlas [46]. Combined with a variety of other residues, such as serine and asparagin, the DH motif frequently forms catalytic sites. Fig. 11 shows four different protein structures with an identified DH motif. LAND is therefore able to identify biologically-relevant motifs in extremely large feature spaces and can be readily used to speed up structure-based models in computational biology. It could also enhance data exploration, e.g., via a recursive study of all DH enzymes, which would result in subtyping of catalytic sites. While identification of structural motifs is not novel in computational biology [47], scaling up such methods to extremely high dimensions is an important new step in the field.

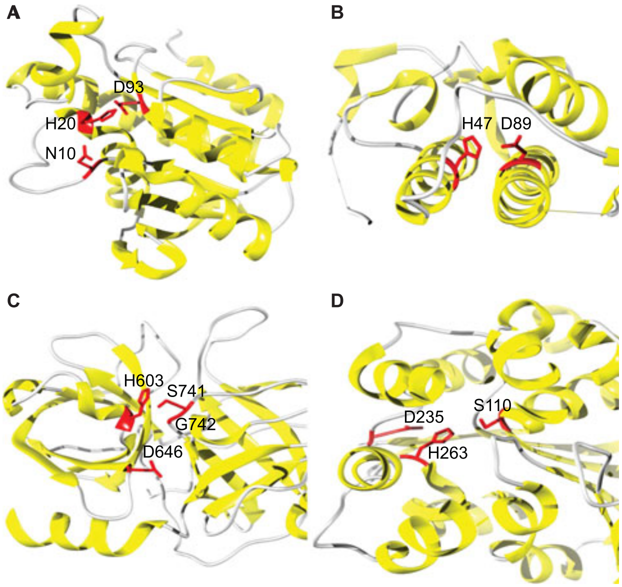


Fig. 11. Structural signatures of enzymes. Illustration of four distinct protein structures from PDB containing the DH structural motif. Catalytic residues, shown in red, were experimentally determined and extracted from Catalytic Site Atlas. (A) Peptidyl-tRNA hydrolase (chain A of PDB entry 2pth) where residues N10, H20 and D93 form a catalytic triad. (B) Basic phospholipase A2 piratoxin-3 (chain A of PDB entry 1gmz) with catalytic residues H47 and D89. (C) Plasminogen (chain A of PDB entry 1qrz) with catalytic residues H603, D646, S741 and G742. (D) 2-hydroxy-6-oxo-6-phenylhexa-2,4-dienoate hydrolase (chain A of PDB entry 1c4x) where residues S110, D236, and H263 form a catalytic triad.

Since LAND can naturally handle structured output information such as link and multi-label data through the output kernel L, applications to structured output data are an interesting direction of future research. Another possible future work would be handle missing values. Finally, using LAND to find a new scientific discovery would be an interesting future work.

Acknowledgements

MY is supported by the JST PRESTO program JPMJPR165A and partly supported by MEXT KAKENHI 16K16114. JT is supported by the National Science Foundation (NSF) under grant number IIS-1714741 and IIS-1715940. HM is supported by MEXT KAKENHI 16H02868, ACCEL JST, and FiDiPro Tekus. PR is supported by the National Science Foundation DBI-1458477.

References

[1] V. Marx, "Biology: The big challenges of big data," *Nature*, vol. 498, no. 7453, pp. 255–260, 2013.

[2] Y. Li and L. Chen, "Big biological data: Challenges and opportunities," *Genomics, Proteomics Bioinf.*, vol. 12, no. 5, pp. 187–189, 2014.

[3] P. R. Burton, et al., "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls," *Nature*, vol. 447, no. 7145, pp. 661–678, 2007.

[4] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[5] Y. Saeys, I. Inza, and P. Larrañaga, “A review of feature selection techniques in bioinformatics,” *Bioinf.*, vol. 23, no. 19, pp. 2507–2517, 2007.

[6] D. Donoho and J. Jin, “Higher criticism thresholding: Optimal feature selection when useful features are rare and weak,” *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 39, pp. 14 790–14 795, 2008.

[7] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1237, Aug. 2005.

[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *Ann. Stat.*, vol. 32, no. 2, pp. 407–499, 2004.

[9] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, “Measuring statistical dependence with Hilbert-Schmidt norms,” in *Proc. 16th Int. Conf. Algorithmic Learn. Theory*, 2005, pp. 63–77.

[10] M. Yamada, W. Jitkrittum, L. Sigal, E. P. Xing, and M. Sugiyama, “High-dimensional feature selection by feature-wise kernelized lasso,” *Neural Comput.*, vol. 26, no. 1, pp. 185–207, 2014.

[11] K. Balasubramanian, B. Sriperumbudur, and G. Lebanon, “Ultrahigh dimensional feature screening via RKHS embeddings,” in *Proc. 16th Int. Conf. Artif. Intell. Stat.*, 2013, pp. 126–134.

[12] L. Yu and H. Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution,” in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 856–863.

[13] B. Senliol, G. Gulgezen, L. Yu, and Z. Cataltepe, “Fast correlation based filter (FCBF) with a different search strategy,” in *Proc. 23rd Int. Symp. Comput. Inf. Sci.*, 2008, pp. 1–4.

[14] I. Rodriguez-Lujan, R. Huerta, C. Elkan, and C. S. Cruz, “Quadratic programming feature selection,” *J. Mach. Learn. Res.*, vol. 11, pp. 1491–1516, 2010.

[15] X. V. Nguyen, J. Chan, S. Romano, and J. Bailey, “Effective global approaches for mutual information based feature selection,” in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 512–521.

[16] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt, “Feature selection via dependence maximization,” *J. Mach. Learn. Res.*, vol. 13, pp. 1393–1434, 2012.

[17] M. Masaeli, G. Fung, and J. G. Dy, “From transformation-based dimensionality reduction to feature selection,” in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 751–758.

[18] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed. Berlin, New York: Springer-Verlag, 2003.

[19] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. R. Stat. Soc., Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[20] Z. Zhao, L. Wang, and H. Liu, “Efficient spectral feature selection with minimum redundancy,” in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 673–678.

[21] M. Tan, I. W. Tsang, and L. Wang, “Towards ultrahigh dimensional feature selection for big data,” *J. Mach. Learn. Res.*, vol. 15, pp. 1371–1429, 2014.

[22] Y. Sun, J. Yao, and S. Goodison, “Feature selection for nonlinear regression and its application to cancer research,” in *Proc. SIAM Int. Conf. Data Min.*, 2015, pp. 73–81.

[23] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman, “Sparse additive models,” *J. R. Stat. Soc.: Series B*, vol. 71, no. 5, pp. 1009–1030, 2009.

[24] H. Liu, J. Lafferty, and L. Wasserman, “Nonparametric regression and classification with joint sparsity constraints,” in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 969–976.

[25] G. Raskutti, M. Wainwright, and B. Yu, “Minimax-optimal rates for sparse additive models over kernel classes via convex programming,” *J. Mach. Learn. Res.*, vol. 13, pp. 389–427, 2012.

[26] T. Suzuki and M. Sugiyama, “Fast learning rate of multiple kernel learning: Trade-off between sparsity and smoothness,” *Ann. Stat.*, vol. 41, no. 3, pp. 1381–1405, 2013.

[27] F. R. Bach, “Exploring large feature spaces with hierarchical multiple kernel learning,” in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 105–112.

[28] P. Jawanpuria, J. S. Nath, and G. Ramakrishnan, “Generalized hierarchical kernel learning,” *J. Mach. Learn. Res.*, vol. 16, pp. 617–652, 2015.

[29] D. He, I. Rish, and L. Parida, “Transductive HSIC lasso,” in *Proc. SIAM Int. Conf. Data Mining*, 2014, pp. 154–162.

[30] C. Cortes, M. Mohri, and A. Rostamizadeh, “Algorithms for learning kernels based on centered alignment,” *J. Mach. Learn. Res.*, vol. 13, pp. 795–828, 2012.

[31] B. Schölkopf and A. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.

[32] A. P. Bradley, “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.

[33] M. Yamada, M. Sugiyama, and T. Matsui, “Semi-supervised speaker identification under covariate shift,” *Signal Process.*, vol. 90, no. 8, pp. 2353–2361, 2010.

[34] T. E. Scheetz, et al., “Regulation of gene expression in the mammalian eye and its relevance to eye disease,” *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 39, pp. 14 429–14 434, 2006.

[35] J. Huang, J. L. Horowitz, and F. Wei, “Variable selection in nonparametric additive models,” *Ann. Stat.*, vol. 38, no. 4, 2010, Art. no. 2282.

[36] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Ann. Stat.*, vol. 29, pp. 1189–1232, 2001.

[37] S. A. Danziger, et al., “Predicting positive p53 cancer rescue regions using most informative positive (MIP) active learning,” *PLoS Comput. Biol.*, vol. 5, no. 9, 2009, Art. no. e1000498.

[38] The Cancer GenomeAtlas, “The molecular taxonomy of primary prostate cancer,” *Cell.*, vol. 163, no. 4, pp. 1011–1025, 2015.

[39] P. Dao, K. Wang, C. Collins, M. Ester, A. Lapuk, and S. C. Sahinalp, “Optimally discriminative subnetwork markers predict response to chemotherapy,” *Bioinf.*, vol. 27, no. 13, pp. i205–i213, 2011.

[40] D. Szklarczyk, et al., “STRING v10: Protein-protein interaction networks, integrated over the tree of life,” *Nucleic Acids Res.*, vol. 43, no. D1, pp. D447–D452, 2014.

[41] H. Berman, et al., “The Protein Data Bank,” *Nucleic Acids Res.*, vol. 28, no. 1, pp. 235–242, 2000.

[42] J. Lugo-Martinez and P. Radivojac, “Generalized graphlet kernels for probabilistic inference in sparse graphs,” *Netw. Sci.*, vol. 2, no. 2, pp. 254–276, 2014.

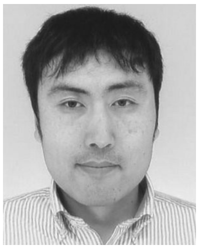
[43] N. Przulj, D. G. Corneil, and I. Jurisica, “Modeling interactome: Scale-free or geometric?” *Bioinf.*, vol. 20, no. 18, pp. 3508–3515, 2004.

[44] N. Przulj, “Biological network comparison using graphlet degree distribution,” *Bioinf.*, vol. 23, no. 2, pp. e177–e183, 2007.

[45] V. Vacic, L. M. Iakoucheva, S. Lonardi, and P. Radivojac, “Graphlet kernels for prediction of functional residues in protein structures,” *J. Comput. Biol.*, vol. 17, no. 1, pp. 55–72, 2010.

[46] C. T. Porter, G. J. Bartlett, and J. M. Thornton, “The Catalytic Site Atlas: A resource of catalytic sites and residues identified in enzymes using structural data,” *Nucleic Acids Res.*, vol. 32, pp. D129–133, 2004.

[47] F. Xin and P. Radivojac, “Computational methods for identification of functional residues in protein structures,” *Curr. Protein Pept. Sci.*, vol. 12, no. 6, pp. 456–469, 2011.



Makoto Yamada received the MS degree in electrical engineering from Colorado State University, Fort Collins, in 2005 and the PhD degree in statistical science from The Graduate University for Advanced Studies (SOKENDAI, The Institute of Statistical Mathematics), Tokyo, in 2010. He has held positions as a systems engineer for the Hitachi Corporation from 2005 to 2007, as a researcher with the Yamaha Corporation from 2007 to 2010, as a postdoctoral fellow with the Tokyo Institute of Technology from 2010 to 2012, as a research associate with NTT Communication Science Laboratories from 2012 to 2013, as a research scientist with Yahoo Labs from 2013 to 2015, and as an assistant professor with Kyoto University from 2015 to 2017. Currently, he is a unit leader at RIKEN AIP. His research interests include machine learning and its application to natural language processing, signal processing, and computer vision.

Jiliang Tang received the BS and MS degrees from the Beijing Institute of Technology, in 2008 and 2010, respectively, and the PhD degree in computer science from Arizona State University, in 2015. He is an assistant professor of computer science and engineering, Michigan State University. His research interests include trust/distrust computing, signed network analysis, social computing, and data mining for social goods. He was awarded the Best Paper Award of SIGKDD2016 and the Runner Up of SIGKDD Dissertation Award 2015. He was the poster chair of SIGKDD2016 and serves as a regular journal reviewer and on numerous conference program committees. He co-presented three tutorials in KDD2014, WWW2014, and Recsys2014, and has published innovative works in highly ranked journals and top conference proceedings that have received extensive coverage in the media.

Jose Lugo-Martinez received dual BS degrees in computer science and mathematics from the University of Puerto Rico-Rio Piedras and the MS degree in computer science from the University of California-San Diego. He is working toward the PhD degree in computer science at Indiana University. His research interests include machine learning, data and text mining, computational biology, and structural bioinformatics. The focus of his doctoral research has been the development of robust kernel methods for learning and inference on noisy and complex network data. In particular, he develops computational approaches towards understanding protein function and how disruption of protein function leads to disease.

Ermin Hodzic is a PhD candidate in Computing Science, Computational Biology, at the Simon Fraser University. He obtained the BS degree in Theoretical Computer Science at University of Sarajevo in 2012, and MS degree in Computational Biology, Computing Science from Simon Fraser University in 2014. His research interests include algorithms in systems biology and the development of combinatorial optimization methods applied to cancer study and precision medicine, in particular revolving around cancer drivers.

Raunak Shrestha received BTech degree in biotechnology from Kathmandu University, in 2009. He is working toward the PhD degree in bioinformatics at the University of British Columbia and is affiliated to the Vancouver Prostate Centre. His research interests include development of graph-based algorithms, machine learning techniques to integrate high-throughput multi-omics data and its applications to study cancer and enable precision oncology. His works have been highlighted in many peer-reviewed publications and top-ranked conferences in bioinformatics including RECOMB2014.

Avishek Saha received the PhD degree in computer science from the School of Computing, University of Utah, in 2012, where he focused on machine learning. He was a research scientist at Yahoo Labs. His research is focused on the theory and algorithms for transfer learning, and applications in ads.

Hua Ouyang received the MPhil degree from the Chinese University of Hong Kong, in 2007, where he worked on large scale multi-modal classification for human speech and computer vision and the PhD degree in computer science from the School of Computational Science and Engineering, College of Computing, Georgia Tech, in 2013, where he focused on designing and analyzing large scale machine learning, stochastic optimization, computational geometry, information retrieval, and recommender systems. He is a senior research engineer at Apple, leading Siri Search Relevance. He was a research scientist at Yahoo Labs. His research is focused on the theory and algorithms for scalable machine learning, and applications in search and recommendation. He worked as an intern at IBM T. J. Watson Research Center, Hawthorne, New York, in 2010, where he worked on large-scale image retrieval and object categorization. He was the recipient of the 2010 best student paper award from the Statistical Computing Section, American Statistical Association. He is a member of the IEEE.

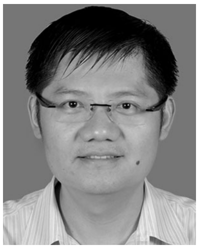
Dawei Yin received BS degree from Shandong University, in 2006, and the MS and PhD degrees from Lehigh University, in 2006 and 2010, respectively. From 2007 to 2008, he was a MPhil student at The University of Hong Kong. He is a director of research with JD.com, leading personalization science team and working on recommender system. Prior to joining JD.com, he was a senior research manager at Yahoo Labs, leading the relevance science team and in charge of core search relevance of Yahoo search. His research interests include data mining, applied machine learning, information retrieval, and recommender system. He published more than 30 research papers in premium conferences, journals, and won the WSDM2016 Best Paper Award and KDD2016 Best Paper Award.

Hiroshi Mamitsuka received the BS degree in biophysics and biochemistry, the ME degree in information engineering, and the PhD degree in information sciences from the University of Tokyo, Tokyo, Japan, in 1988, 1991, and 1999, respectively. He is a professor with Kyoto University and a FiDiPro professor in the Department of Computer Science, Aalto University, Finland. He is involved in research on machine learning, data mining, and bioinformatics. His current research interests include mining from graphs and networks in biology and chemistry.

Cenk Sahinalp received the BSc degree in electrical engineering from Bilkent University and the PhD degree in computer science from the University of Maryland, College Park. He is a professor of computer science and the co-director of the Center for Genomics and Bioinformatics, Indiana University, Bloomington, on leave from the School of Computing Science, Simon Fraser University and the Vancouver Prostate Centre. He has co-authored more than 120 conference and journal publications in some of the leading venues of algorithms, bioinformatics, genomics, and cancer biology. His current research interests include algorithmic infrastructure for bioinformatics research and integrative cancer bioinformatics.

Predrag Radivojac received the BS degree from the University of Novi Sad, in 1994, the MS degree from the University of Belgrade, in 1997, both in electrical engineering, and the PhD degree in computer and information sciences from Temple University under the direction of Prof. Zoran Obradovic, in 2003. He is a professor of computer science with Indiana University. In 2004, he held a post-doctoral position in Prof. Keith Dunker's lab at Indiana University's School of Medicine, after which he joined Indiana University, Bloomington as a faculty member. His research interests include computational biology, biomedical informatics, and machine learning. He received the US National Science Foundation CAREER Award, in 2007 dedicated to understanding protein post-translational modifications. Currently, he is an editorial board member for the journal *Bioinformatics*, an associate editor for *PLoS Computational Biology*, and serves on the Board of Directors of the International Society for Computational Biology (ISCB).

Filippo Menczer received the Laurea degree in physics from the Sapienza University of Rome and the PhD degree in computer science and cognitive science from the University of California, San Diego. He is a professor of informatics and computer science with Indiana University, Bloomington. He currently serves as a director of the Center for Complex Networks and Systems Research and on the senior leadership team of the IU Network Science Institute. He is a fellow of the Institute for Scientific Interchange Foundation in Torino, Italy, a senior research fellow of The Kinsey Institute, and an ACM distinguished scientist. His research interests include web science, social networks, social media, social computation, web mining, distributed and intelligent web applications, and modeling of complex information networks.



Yi Chang is a professor at Jilin University in China. He has broad research interests on information retrieval, data mining, machine learning, natural language processing, and artificial intelligence. He has published more than 100 research papers in premium conferences or journals, and he is an associate editor of IEEE TKDE.