# Consistent and Efficient Nonparametric Different-Feature Selection

**Satoshi Hara**
National Institute of Informatics, Japan
JST, ERATO, Kawarabayashi Large Graph Project
satohara@nii.ac.jp

**Takayuki Katsuki** **Hiroki Yanagisawa**
IBM Research – Tokyo, Japan
kats@jp.ibm.com, yanagis@jp.ibm.com

**Takafumi Ono**
University of Bristol, UK
takafumi.ono@bristol.ac.uk

**Ryo Okamoto** **Shigeki Takeuchi**
Kyoto University, Japan
okamoto.ryo.4w@kyoto-u.ac.jp, takeuchi@kuee.kyoto-u.ac.jp

## Abstract

Two-sample feature selection is a ubiquitous problem in both scientific and engineering studies. We propose a feature selection method to find features that describe a difference in two probability distributions. The proposed method is nonparametric and does not assume any specific parametric models on data distributions. We show that the proposed method is computationally efficient and does not require any extra computation for model selection. Moreover, we prove that the proposed method provides a consistent estimator of features under mild conditions. Our experimental results show that the proposed method outperforms the current method with regard to both accuracy and computation time.

## 1 Introduction

Two-sample feature selection is a task of finding features with distribution differences between two datasets. Feature selection helps us understand what causes differences between the datasets, which is a fundamental question in both scientific and engineering studies. Important example tasks include the two-sample test [1, 2] and anomaly detection [3, 4]. For example, in gene expression data analysis, a two-sample test-based approach allows us to find genes that are specific to some subtypes [5]. In the anomaly detection context, one can find causes of an error by localizing features that behave differently between the datasets sampled before and after the occurrence of the error [6].

In this paper, we focus on finding features that describe a difference between two probability distributions. Suppose we have i.i.d. samples from the probability distributions $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ of sizes $N$ and $M$, respectively, where $\boldsymbol{x} \in \mathbb{R}^D$ is a $D$-dimensional feature. Using these samples, we aim to find a subset of features $S \subseteq \{1, 2, \ldots, D\}$ in which the two distributions do not match. Intuitively, we expect that $p(\boldsymbol{x}_S) \neq q(\boldsymbol{x}_S)$ and $p(\boldsymbol{x}_{S^c}) = q(\boldsymbol{x}_{S^c})$ hold, where $\boldsymbol{x}_S$ and $\boldsymbol{x}_{S^c}$ denote subsets of a random variable $\boldsymbol{x}$ specified by the set $S$ and its complement $S^c$, respectively. We refer to the problem as *different-feature selection*.

There have been several studies on different-feature selection in the two sample test and the anomaly detection contexts. In the two sample test context, Benjamini and Hochberg [1] proposed comparing each signle feature using statistical tests, and then adjusting the false discovery rate. In the anomaly detection context in which the objective is to find features with anomalies, the Mahalanobis-Taguchi System (MT) [3] is one of the most classical method. The MT models both $p$ and $q$ as Gaussian and then finds features with a different mean or covariance. Following MT, several lines of research focused on different-feature selection under the Gaussian setting. Hirose et al. [7] proposed using the change in inter-sensor correlations to find features with distribution changes. Jiang et al. [8] proposed a PCA-based method. Idé et al. [6, 9] used the changes in the correlation and partial correlation. In our previous study [4], we proposed an algorithm with a consistency guarantee.

Proceedings of the $20^{th}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

Table 1: Comparison of nonparametric different-feature selection methods.

| | SPARDA [5]<br>(w/ projected gradient) | SPARDA [5]<br>(w/ relax and tighten) | Proposed Method |
|---|---|---|---|
| Computation Speed | **Fast** | Slow | **Fast** |
| Cross Validation | Necessary | Necessary | **Not Necessary** |
| Optimality | Local optimal | **Nearly global optimal in practice** | **Nearly global optimal in practice** |

Recently, Mueller and Jaakkola [5] proposed a new different-feature selection method called SPARDA. SPARDA finds a feature set $S$ by searching a subspace with the maximum distribution difference by solving a nonconvex problem. In particular, Mueller and Jaakkola [5] used a nonparametric metric called Wasserstein distance [10] to measure a difference between the distributions. Because the Wasserstein distance is nonparametric, SPARDA does not assume any specific parametric models on $p$ and $q$. This property contrasts with the MT and its variants where the Gaussian distribution is used. This nonparametric nature of SPARDA is favorable in practice because we usually do not know the data distribution models, and they can be non-Gaussian in many cases. Mueller and Jaakkola [5] also proved that SPARDA provides a consistent estimator of the feature subset $S$. The major difficulty with SPARDA, however, is solving the nonconvex optimization problem. Mueller and Jaakkola [5] proposed a *relax and tighten* procedure that can find nearly global optima; however, this procedure incurs high computational complexity. It solves a semidefinite program at every iteration, each of which runs in $O(D^3NM)$ time. Therefore, applying the relax and tighten procedure to large datasets is difficult. Projected gradient ascent is a faster alternative method that runs in $O(D(N+M)+N\log N + M\log M)$ time per iteration. However, it is easily trapped by local optima, as we will see in our experiments. Note that in practice, the computation time of these methods is further increased by the need for cross validation for model selection; SPARDA needs to choose an optimal regularization parameter.

This literature survey reveals the limitations of existing different-feature selection methods. The Gaussian-based methods have limited applicability due to the restrictive Gaussian assumption, whereas the nonparametric SPARDA approach has computational difficulty. Therefore, a computationally efficient different-feature selection method with a less restrictive assumption is required.

Our major contributions are twofold. First, we propose a simple nonparametric method for different-feature selection. The proposed method does not assume any specific parametric models on $p$ and $q$, and its time complexity is only $O(D^2(N+M)\log NM)$ on average. Moreover, the proposed method does not require optimization of any regularization parameters; thus, it does not require any extra computation for model selection. We formulate the problem as a sparsest $k$-subgraph problem [11] using the estimated KL-divergence. Although the problem is NP-hard in general, we derive a nearly global optimal solution using a greedy method. Table 1 summarizes the properties of the proposed method and SPARDA.

The second contribution is consistency theorems for the proposed method. The consistency of the different-feature selection method was first proved in our previous study [4, Corollary 1] under the Gaussian setting. Mueller and Jaakkola [5, Theorem 4] proved the consistency of SPARDA without assuming any specific parametric models. We prove the consistency of the proposed method under mild assumptions on the distributions $p$ and $q$. Our consistency guarantee requires conditions only on the KL-divergence between the data distributions but not on their distribution models.

Our experimental results confirmed the high accuracy and computational efficiency of the proposed method for both synthetic and real-world data. We found that the proposed nonparametric method can detect a complex distribution difference effectively and outperforms Gaussian-based methods. We also compared the proposed method and SPARDA with projected gradient ascent for both accuracy and runtime. The results showed that the proposed method attained higher accuracy for many cases. We conjecture that SPARDA tended to be trapped by local optima, while the proposed method could find a nearly global optima using the greedy method. We also observed that the speed of the proposed method was comparable to or even several times faster than SPARDA.

## 2 Problem Definition

**Notation:** Let $[D] := \{1, 2, \ldots, D\}$ for $D \in \mathbb{N}$. For a vector $\boldsymbol{x} \in \mathbb{R}^D$, $x_d$ is its $d$-th component, and for a matrix $L \in \mathbb{R}^{D\times D}$, $L_{dd'}$ is its $(d, d')$-th component. For a set $S \subseteq [D]$, $S^{\mathrm{c}} := [D] \setminus S$ is its complement. For a vector $\boldsymbol{x}$ and a set $S \subseteq [D]$, $\boldsymbol{x}_S := \{x_d \mid d \in S\}$ is a feature subset. $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ denotes the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\Sigma$. $\mathbf{0}_D$ denotes the $D$-dimensional vector with all entries equal to zero.

We now define the different-feature selection problem we consider in this paper. Let $\boldsymbol{x} := (x_1, x_2, \ldots, x_D)^\top \in \mathbb{R}^D$ be a $D$-dimensional feature.

We aim to find features in which the distributions do not match between two distributions. That is, for a subset $S \subseteq [D]$, we expect that there is a distribution difference in the $d$-th feature $x_d$ when $d \in S$, while there is no distribution difference in the $d'$-th feature $x_{d'}$ when $d' \notin S$. We formalize the problem as follows.

**Problem 1 (Different-Feature Selection)** *Given i.i.d. samples $\mathcal{P} = \{\boldsymbol{y}^{(n)}\}_{n=1}^{N} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x})$ and $\mathcal{Q} = \{\boldsymbol{z}^{(m)}\}_{m=1}^{M} \overset{\text{i.i.d.}}{\sim} q(\boldsymbol{x})$, identify the set $S \subseteq [D]$ that satisfies*

$$p(\boldsymbol{x}_{S^{\mathrm{c}}}) = q(\boldsymbol{x}_{S^{\mathrm{c}}}), \tag{1}$$

$$p(\boldsymbol{x}_{S^{\mathrm{c}} \cup \{d\}}) \neq q(\boldsymbol{x}_{S^{\mathrm{c}} \cup \{d\}}),\ \forall d \in S. \tag{2}$$

Conditions (1) and (2) require that the distributions match on the feature subset $S^{\mathrm{c}}$, and the equation does not hold when the feature $d \in S$ is removed from $S$ and added to $S^{\mathrm{c}}$

We note that Problem 1 is a generalization of a common feature selection problem for binary classification. Whereas existing methods, such as Lasso logistic regression [12], search for discriminative features between the two classes, in Problem 1, we also search for non-discriminative features with distribution differences (e.g., features with variance changes).

# 3 Proposed Nonparametric Method

We propose a simple nonparametric method for different-feature selection that satisfies two requirements, i.e., less restrictive assumption and computational efficiency. Specifically, we derive a computationally efficient algorithm by focusing only on the difference of marginal distributions on the pair of features. Moreover, the proposed method can capture the difference on higher-order moments of distributions, which is overlooked by the Gaussian-based methods. We present the algorithm in this section, and provide two important properties in the next section.

The proposed method consists of KL-divergence estimation and solving a sparsest $k$-subgraph problem [11]. In the proposed method, we use the univariate and bivariate KL-divergences $\mathrm{KL}[p(x_d)||q(x_d)]$ and $\mathrm{KL}[p(x_d, x_{d'})||q(x_d, x_{d'})]$. Here, $p(x_d)$ and $q(x_d)$ are univariate distributions on $p$ and $q$, and $p(x_d, x_{d'})$ and $q(x_d, x_{d'})$ are bivariate distributions on $p$ and $q$, respectively. Note that $p(x_d) = q(x_d)$ is required for $d \in S^{\mathrm{c}}$ from Condition (1). Thus, we have $\mathrm{KL}[p(x_d)||q(x_d)] = 0$ for $d \in S^{\mathrm{c}}$. Similarly, we have $\mathrm{KL}[p(x_d, x_{d'})||q(x_d, x_{d'})] = 0$ for $d, d' \in S^{\mathrm{c}}$. In contrast, we may have $\mathrm{KL}[p(x_d, x_{d'})||q(x_d, x_{d'})] > 0$ for some $d \in S$. This is because the removal of $d$ from $S$ violates the equality, as in Condition (2), which indicates that there may exist $d' \in [D] \setminus \{d\}$ with the distribution difference $p(x_d, x_{d'}) \neq q(x_d, x_{d'})$. By using these properties on each univariate and bivariate distribution, we estimate the set $S$.

In the proposed method, we first compute a *KL-divergence matrix* between $\mathcal{P}$ and $\mathcal{Q}$. We then estimate the set $S$ by solving the sparsest $k$-subgraph problem [11] on the KL-divergence matrix.

## 3.1 Step 1: Compute KL-divergence Matrix

We define the KL-divergence matrix $L \in \mathbb{R}^{D \times D}$ as $L_{dd} = \mathrm{KL}[p(x_d)||q(x_d)]$ for $d \in [D]$ and $L_{dd'} = \mathrm{KL}[p(x_d, x_{d'})||q(x_d, x_{d'})]$ for $d, d' \in [D], d \neq d'$. In practice, we use the estimated KL-divergence using the nearest-neighbor-based method [13]: for $T \subseteq [D]$, $\widehat{\mathrm{KL}}[p(\boldsymbol{x}_T)||q(\boldsymbol{x}_T)] := \frac{|T|}{N} \sum_{n=1}^{N} \log \frac{\nu(\boldsymbol{y}^{(n)};T)}{\rho(\boldsymbol{y}^{(n)};T)} + \log \frac{M}{N-1}$, where $\rho(\boldsymbol{y}^{(n)}; T) := \min_{\boldsymbol{y}' \in \mathcal{P} \setminus \{\boldsymbol{y}^{(n)}\}} \|\boldsymbol{y}_T^{(n)} - \boldsymbol{y}'_T\|$ and $\nu(\boldsymbol{y}^{(n)}; T) := \min_{\boldsymbol{z} \in \mathcal{Q}} \|\boldsymbol{y}_T^{(n)} - \boldsymbol{z}_T\|$. We denote the estimated KL-divergence matrix as $\hat{L}$ hereafter. The nearest-neighbor-based estimator is asymptotically unbiased and consistent under a regularity condition [13, Theorem 1, 2]. Note that the estimated KL-divergence can be negative; nevertheless, the method and the theoretical analysis are valid. Because Step 1 is composed of $\frac{D(D+1)}{2}$ independent computations, it can be parallelized easily.

## 3.2 Step 2: Solve Sparsest $k$-subgraph Problem

Estimation of the set $S$ amounts to finding a submatrix of $\hat{L}$ whose entries are close to zero. This is because $\hat{L}_{dd'} \approx 0$ is expected for $d, d' \in S^{\mathrm{c}}$ from Condition (1), while $\hat{L}_{dd'} > 0$ is expected for some $d \in S$ from Condition (2). Such a set $S$ can be derived by solving the following problem:

$$\hat{S} = \operatorname*{argmin}_{S \subseteq [D]} \textstyle\sum_{d, d' \in S^{\mathrm{c}}} \hat{L}_{dd'} = f(S), \text{ s.t. } |S| = \alpha, \tag{3}$$

where $\alpha$ is the number of features with distribution differences. The problem (3) is equivalent to the sparsest $k$-subgraph problem [4] with $k = D - \alpha$, and is NP-hard in general [11]. The exact solution can be derived using state-of-the-art solvers such as IBM ILOG CPLEX although it may take exponential time.

In practice, we can use a greedy method to derive a pragmatic solution in reasonable time, as shown in our previous study [4, Algorithm 1]. One difficulty with the greedy method, however, is that $\alpha$ is unknown in most cases. Therefore, we propose a new heuristic algorithm (Algorithm 1) to avoid specifying the number $\alpha$. With the algorithm, we score the feature $x_d$

**Algorithm 1** Greedy Scoring Method

```
Input: KL-divergence matrix $\hat{L} \in \mathbb{R}^{D \times D}$
Output: Score vector $\hat{\boldsymbol{s}} \in \mathbb{R}^{D}$
  Define $f(S) := \sum_{\delta, \delta' \in S^{\mathrm{c}}} \hat{L}_{\delta\delta'}$
  Let $\hat{S} \leftarrow \emptyset$, $\hat{\boldsymbol{s}} \leftarrow \mathbf{0}_D$
  for $i = 1$ to $D$ do
    $d \leftarrow \mathrm{argmin}_{d' \in \hat{S}^{\mathrm{c}}} f(\hat{S} \cup \{d'\})$
    $\hat{s}_d \leftarrow (f(\hat{S}) - f(\hat{S} \cup \{d\})) \,/\, (D - i + 1)$
    $\hat{S} \leftarrow \hat{S} \cup \{d\}$
  end for
```

based on the normalized change in the function value $f$ when an element $d$ is added to $\hat{S}$. If the addition of $d$ to $\hat{S}$ significantly reduces the function value, we can conjecture that there is a distribution difference in the feature $x_d$. Formally, we estimate the set $S$ by $\hat{S}_t := \{d \mid \hat{s}_d > t\}$ by applying a threshold $t$ to the score $\hat{\boldsymbol{s}}$ derived from Algorithm 1. This procedure is more practical than the original greedy method because $\alpha$ does not need to be specified explicitly. The threshold $t$ can be determined, for instance, by a visual inspection of the score bar chart. One can also use a convex relaxation method [4] to solve the problem (3). One can derive a sparse solution that does not require specifying a threshold at a cost of computation time.

# 4 Theoretical Analysis

## 4.1 Computational Complexity

Here, we show that the proposed method runs in $O(D^2(N+M)\log NM)$ average time. In the proposed method, Step 1 runs in $O(D^2(N+M)\log NM)$ average time, and Step 2 runs in $O(D^2)$ time which is negligible compared to that of Step 1.

In Step 1, for the computation of $\widehat{\mathrm{KL}}[p(\boldsymbol{x}_T)||q(\boldsymbol{x}_T)]$, we search for the nearest-neighbors for each of the $N$ points $\boldsymbol{y}^{(n)}$ from the sets $\mathcal{P}$ and $\mathcal{Q}$. Note that we can search for the nearest-neighbors efficiently using a k-d tree [14]. We require $O(N\log N + M\log M)$ time to construct k-d trees for both $\mathcal{P}$ and $\mathcal{Q}$ [14]. For the nearest-neighbor search, although we may require $O(N+M)$ time in the worst case, the average time is $O(\log NM)$ in practice. Thus, we have an overall search complexity for all $N$ points as $O(N\log NM)$ on average. Because we estimate the KL-divergence for each of $\frac{D(D+1)}{2}$ components in $\hat{L}$, we have the average time complexity as $O(D^2(N+M)\log NM)$.

Step 2 has less time complexity compared to Step 1. Algorithm 1 runs in $O(D^2)$ time by using a book keeping. In book keeping, we maintain $\boldsymbol{a} \in \mathbb{R}^D$ such that $a_{d'} := \sum_{\delta \in \hat{S}^{\mathrm{c}}} \hat{L}_{d'\delta}$ for every $d' \in \hat{S}^{\mathrm{c}}$. Then, in every iteration, the value of $f(\hat{S} \cup \{d\})$ can be computed as $f(\hat{S} \cup \{d\}) = f(\hat{S}) - 2a_d + \hat{L}_{dd}$ which is $O(1)$ time for every $d \in \hat{S}^{\mathrm{c}}$. Thus, the argmin operation can be computed in $O(D)$ time. We then update $\boldsymbol{a}$ by $a_{d'} \leftarrow a_{d'} - \hat{L}_{d'd}$ when an update $\hat{S} \leftarrow \hat{S} \cup \{d\}$ is executed, which is also $O(D)$ time. Hence, one iteration in Algorithm 1 runs in $O(D)$ time, and the overall time complexity is $O(D^2)$.

## 4.2 Consistency Guarantee

We give a consistency guarantee for the estimated set $\hat{S}$ derived by solving the problem (3): $\lim_{N,M\to\infty} P(\hat{S} \neq S) = 0$.

This guarantee is based on the following assumption about the distributions $p$ and $q$.

**Assumption 1 (Regularity [13])** *There exists an $\epsilon > 0$ such that the following conditions hold for all $T \subseteq [D]$ with $|T| \leq 2$:*

$$\int |\log p(\boldsymbol{x}_T)|^{2+\epsilon} p(\boldsymbol{x}_T) d\boldsymbol{x}_T < \infty,$$
$$\int |\log \|\boldsymbol{x}_T - \boldsymbol{y}_T\||^{2+\epsilon} p(\boldsymbol{x}_T) p(\boldsymbol{y}_T) d\boldsymbol{x}_T d\boldsymbol{y}_T < \infty,$$
$$\int |\log q(\boldsymbol{x}_T)|^{2+\epsilon} p(\boldsymbol{x}_T) d\boldsymbol{x}_T < \infty,$$
$$\int |\log \|\boldsymbol{x}_T - \boldsymbol{y}_T\||^{2+\epsilon} p(\boldsymbol{x}_T) q(\boldsymbol{y}_T) d\boldsymbol{x}_T d\boldsymbol{y}_T < \infty.$$

Assumption 1 requires the distributions $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ to decay sufficiently fast, i.e., they are not very heavy-tailed. Common distributions such as Gaussian and Laplacian satisfy this assumption.

The next two theorems indicate that we can derive a consistent estimator of the set $S$ without assuming any specific parametric models on $p$ and $q$. This contrasts with our previous method [4] where consistency is guaranteed under some limited distributions such as Gaussian. All proofs can be found in the supplemental material (Appendix 11).

**Theorem 1 (Necessary Condition)** *Suppose Assumption 1 holds. If $\hat{S}$ is consistent, one of* (N1) *and* (N2) *holds for any $d \in S$:* (N1) $L_{dd} > 0$, (N2) $\exists d' \in [D] \setminus \{d\}$, $L_{dd'} > 0$.

**Theorem 2 (Sufficient Condition)** *$\hat{S}$ is consistent when Assumption 1 and one of* (S1) *and* (S2) *holds for any $d \in S$:* (S1) $L_{dd} > 0$, (S2) $\forall d' \in [D] \setminus \{d\}$, $L_{dd'} > 0$.

Conditions (N1) and (N2) require the distribution difference to be observed on some pair of features. Note that this is not a restrictive condition in practice. Conditions (N1) and (N2) are violated only when the difference appears on the distribution of more than two variables, i.e., $p(x_d, x_{d'}, x_{d''}) \neq q(x_d, x_{d'}, x_{d''})$ holds while $p(\boldsymbol{x}_T) = q(\boldsymbol{x}_T)$ for any $T \subsetneq \{d, d', d''\}$. Intuitively, these cases are negligible in practice as they require the distributions $p$ and $q$ to have very specific structures. The following theorem guarantees that this intuition is correct in the Gaussian case. Indeed, Conditions (N1) and (N2) hold for any distribution differences under Problem 1 with a Gaussian distribution.

**Theorem 3** *When both $p$ and $q$ are Gaussian, one of* (N1) *and* (N2) *holds for any $d \in S$.*

## 5 Experiments

We evaluated the different-feature selection performance of the proposed method both on its accuracy and runtime. We first give illustrative examples with synthetic data that describe the advantage and disadvantage of the proposed method. We then present experimental results on UCI datasets and on a quantum system anomaly detection application. All experiments were conducted using a 16-core VM with an Intel Xeon E312xx, 16GB of RAM, and Ubuntu 15.04.

**Baseline Methods:** We compared the proposed method to four baseline methods. The first three are the Gaussian-based methods MT [3], Idé'09 [6], and Hara'15 [4], and the last one is the nonparametric method SPARDA [5]. See the supplemental material for the details of each method.

**Implementations:** In the experiments, we used the greedy scoring method (Algorithm 1) for the proposed method. The proposed method and Gaussian-based methods were implemented in Python. SPARDA was implemented in C++ based on the MATLAB code `fastSPARDA.m`, which is available on the author's website (`http://www.mit.edu/~jonasm/`). For SPARDA, because the relax and tighten procedure was too slow, we used the projected gradient ascent, which runs in $O(D(N+M) + N \log N + M \log M)$ time per iteration. Because the projected gradient ascent tends to be trapped by local optima, we used five random restarts. We set the regularization parameter candidate for SPARDA to $\{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ and selected the optimal one using 5-fold cross validation.

**Evaluation Metric:** Each method outputs a $D$-dimensional score vector that describes how likely the corresponding feature has changed. We compare the score vector to the ground truth features $S$, and then measure the area under the receiver operating characteristic curve (AUROC). AUROC$=1$ means that the features are correctly identified with high scores.

### 5.1 Illustrative Examples

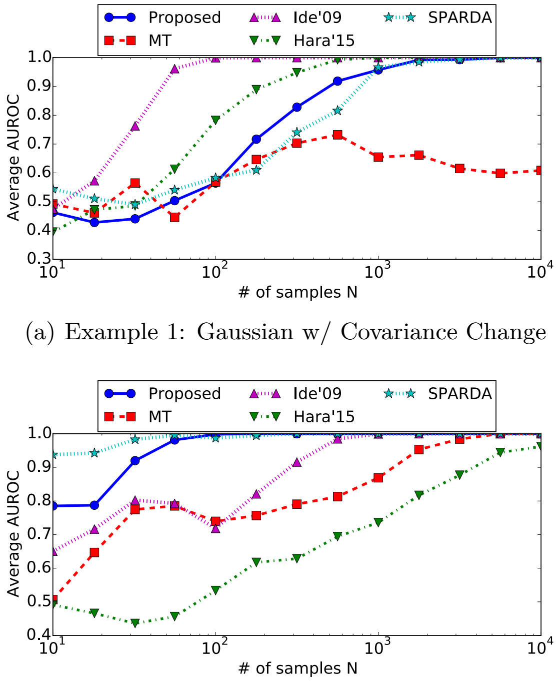
Here, we show the advantage and disadvantage of the proposed method on synthetic experiments. We also present a runtime comparison of the proposed method and SPARDA.

**[Example 1] Gaussian with Covariance Change:** In the first example, we used Gaussian data. We generated synthetic data as follows: Let $\Theta$ be a $20 \times 20$ randomly generated matrix from $\mathcal{N}(0, 1)$. We then computed $\Sigma = \Theta^\top \Theta$ and normalized the diagonal of $\Sigma$ to be one. We then generated 20-dimensional data from the distributions $p(\boldsymbol{x}) = \mathcal{N}(\mathbf{0}_{20}, \Sigma)$ and $q(\boldsymbol{x}) = \mathcal{N}(\mathbf{0}_{20}, \Sigma')$, where $\Sigma'_{11} = 0.49\Sigma_{11} + 0.09\Sigma_{22} + 0.21\Sigma_{12}$, $\Sigma'_{1d} = 0.7\Sigma_{1d} + 0.3\Sigma_{dd}$ for $d \in [20] \setminus \{1\}$, and $\Sigma_{dd'} = \Sigma'_{dd'}$ otherwise. In this setting, $S = \{1\}$ is the solution to Problem 1. We set the numbers of data points in $\mathcal{P}$ and $\mathcal{Q}$ to be equal, i.e., $N = M$. Then, we randomly generated datasets 100 times for several different dataset sizes $N$.

Figure 1(a) shows the average AUROC of each method over 100 random data realizations. Idé'09 and Hara'15 converged to an average AUROC $= 1$ around $N = 10^2$ and $N = 10^3$, respectively. The proposed method attained an average AUROC $= 1$ around $N = 3 \times 10^3$, which is slower than the previous two methods. This shows that the use of the correct parametric model is advantageous in different-feature selection. However, it is noteworthy that the proposed method provided a consistent result with large sample sizes, as implied from Theorem 2. In other words, the proposed method can be an alternative to Gaussian-based methods when there is a sufficiently large number of samples. Note that SPARDA attained a comparable average AUROC.

**[Example 2] Gaussian Mixture with Mixture Rate Change:** In the second example, we used non-Gaussian data to show the advantage of the proposed method. In the example, we generated 20-dimensional data from the Gaussian mixture distributions $p$ and $q$ with different mixture rates on feature $x_1$. Let $p(\boldsymbol{u}) = \mathcal{N}(\mathbf{0}_{20}, \Sigma)$ be a 20-dimensional Gaussian distribution. We defined $p(x_d|u_d) = 0.5\delta(x_d - u_d - 4) + 0.5\delta(x_d - u_d + 4)$ for $d = 1$ and $p(x_d|u_d) = \delta(x_d - u_d)$ otherwise, where $\delta(\cdot)$ is a delta function. We also defined $q(x_d|u_d) = 0.35\delta(x_d - u_d - 4) + 0.35\delta(x_d - u_d + 4) + 0.3\delta(x_d - u_d)$ for $d = 1$ and $q(x_d|u_d) = \delta(x_d - u_d)$ otherwise. In this setting, $S = \{1\}$ is the solution to Problem 1. Note that the change from $p$ to $q$ causes variance change in feature $x_1$; therefore, it can be detected using the Gaussian-based methods.

Figure 1(b) shows the advantage of the proposed method and SPARDA. They attained an average AUROC $= 1$ around $N = 10^2$, which is a significantly fast convergence compared to the Gaussian-based methods. Idé'09 required $N = 10^3$ to attain an average AUROC $= 1$, and MT and Hara'15 required more samples. This indicates that the proposed method and SPARDA can detect the complex distribution difference effectively due to their nonparametric nature. Thus, they performed well with non-Gaussian data where the Gaussian-based methods performed poorly. Note that SPARDA performed better than the proposed method for small sample sizes in this example.



(a) Example 1: Gaussian w/ Covariance Change

(b) Example 2: Gaussian Mixture w/ Rate Change

Figure 1: Comparison of AUROC

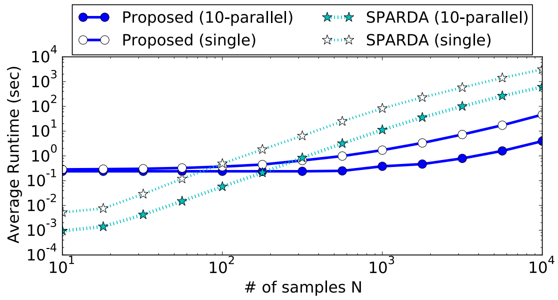


Figure 2: Comparison of runtime for Example 1. Runtimes of single-thread and ten-thread implementations were measured.

**Runtime Comparison:** Figure 2 shows the entire runtime of the proposed method and SPARDA for Example 1. For comparison, we used both single-thread and ten-thread implementations. In the ten-thread implementation, the computation of the matrix $\hat{L}$ was parallelized in the proposed method, while the parameter search with cross validation and random restarts were parallelized in SPARDA.

Table 2: Datasets from the UCI Repository. $D_0$ is number of features and $N_0$ is number of data points. $D$ denotes number of effective features after screening; we removed features taking less than 10 different values. Each dataset is normalized so that mean of each feature is zero and variance is one. CBM dataset is from Coraddu et al. [15].

| | $D_0$ | $D$ | $N_0$ |
|---|---|---|---|
| CASP | 10 | 10 | 45730 |
| CBM | 18 | 13 | 11934 |
| Diagnosis | 48 | 48 | 58509 |
| MiniBooNE | 50 | 50 | 130065 |
| Statlog | 37 | 36 | 6435 |

From Figure 2, we find that the proposed method was significantly faster than SPARDA for large sample sizes. This was because the proposed method has small time complexity and does not require any extra computation for model selection. For $N \geq 10^3$, with both the single-thread and ten-thread implementations, the proposed method was more than 100 times faster than SPARDA. Together with Figure 1(a), the result shows that the proposed method could provide consistent solutions with more than 100 times less runtime. By contrast, SPARDA was computationally advantageous for small sample sizes.

### 5.2 Experiments on UCI Datasets

Here, we present experimental results on five real-world datasets from the UCI repository [16]. The list of the datasets is shown in Table 2. These datasets are non-Gaussian and are, therefore, appropriate for evaluating the effectiveness of the proposed method.

We constructed the datasets $\mathcal{P}$ and $\mathcal{Q}$ from each dataset, each of which consists of randomly chosen $N = M = 1,000$ data points without overlap. For dataset $\mathcal{Q}$, we randomly selected a feature subset $S \subset [D]$ with $|S| = 3$ and modified the distribution of $\boldsymbol{x}_S$. Specifically, for $d \in S$ and $d' \in S^c$, we applied one of the following five changes: (a) Mean Shift $x_d \leftarrow x_d + c$; (b) Variance Change $x_d \leftarrow x_d + c\epsilon$, $\epsilon \sim \mathcal{N}(0, 1)$; (c) Covariance Change $x_d \leftarrow (1-c)x_d + cx_{d'}$; (d) Covariance Change (Conditional) $x_d \leftarrow (1-c)x_d + cx_{d'}$ when $x_{d'} \leq v$; (e) Covariance Change (No Variance Change) $x_d \leftarrow w(1-c)x_d + wcx_{d'}$, where $c \in [0, 1]$ is the parameter that controls the difference level, $v$ is the 25% quantile of $x_{d'}$ in the dataset $\mathcal{Q}$, and $w$ is a scalar factor that maintain the variance of $x_d$ unchanged. Note that these changes affect the mean or covariance of the distribution; thus, they can be detected using the Gaussian-based methods.

Table 3 shows the results of Covariance Change with two difference levels. The results for the other changes were similar to those shown in Table 3, and we moved them to the supplemental material (Appendix 9).

From Table 3, we find three important results that show the effectiveness of the proposed method. The first finding is that the AUROC of the proposed method attained the best average score among the five methods for all cases. Moreover, we observe that there is more than 0.2 improvement of the average AUROCs between the proposed method and Gaussian-

Table 3: UCI Dataset Results: Left – average AUROC ± standard deviation on 20 random data realizations. Proposed (exact) is a referential result with the exact solution of (3) derived using IBM ILOG CPLEX. The highest AUROC among five methods is shown in bold letters. The best results and other results were compared using a t-test (5%), and results that were not rejected are also highlighted; Right – average runtime ± standard deviation of the proposed method and SPARDA with ten-thread parallelization. The smaller runtime is highlighted.

[Covariance Change]

| | | AUROC | | | | | | Runtime (sec) | |
|---|---|---|---|---|---|---|---|---|---|
| | $c$ | Proposed (exact) | Proposed | MT [3] | Idé’09 [6] | Hara’15 [4] | SPARDA [5] | Proposed | SPARDA |
| CASP | .3 | .92 ± .11 | **.93 ± .07** | .61 ± .20 | .86 ± .10 | .84 ± .14 | .75 ± .18 | **0.37 ± 0.07** | 8.51 ± 4.93 |
| | .5 | .98 ± .07 | **.98 ± .03** | .66 ± .18 | .90 ± .06 | .92 ± .11 | .77 ± .19 | **0.36 ± 0.08** | 4.98 ± 3.77 |
| CBM | .3 | .95 ± .09 | **.96 ± .07** | .41 ± .15 | .81 ± .15 | .82 ± .11 | .62 ± .15 | **0.41 ± 0.06** | 19.3 ± 9.74 |
| | .5 | .96 ± .09 | **.98 ± .04** | .45 ± .18 | .82 ± .14 | .84 ± .11 | .70 ± .13 | **0.41 ± 0.04** | 12.7 ± 10.5 |
| Diag nosis | .3 | .90 ± .09 | **.94 ± .06** | .45 ± .16 | .82 ± .13 | .79 ± .13 | .47 ± .14 | **1.35 ± 0.08** | 31.1 ± 54.5 |
| | .5 | .95 ± .08 | **.97 ± .04** | .50 ± .11 | .87 ± .11 | .87 ± .12 | .62 ± .24 | **1.33 ± 0.07** | 24.3 ± 40.2 |
| Mini BooNE | .3 | .74 ± .14 | **.94 ± .05** | .45 ± .13 | .60 ± .16 | .54 ± .13 | .55 ± .19 | **1.64 ± 0.12** | 153 ± 58.8 |
| | .5 | .88 ± .12 | **.98 ± .02** | .44 ± .13 | .65 ± .19 | .58 ± .15 | .56 ± .20 | **1.68 ± 0.10** | 138 ± 57.0 |
| Stat log | .3 | 1.0 ± .00 | **1.0 ± .00** | .42 ± .18 | **1.0 ± .00** | .95 ± .07 | .67 ± .26 | **0.67 ± 0.06** | 10.4 ± 4.32 |
| | .5 | .98 ± .05 | **.98 ± .07** | .27 ± .09 | **1.0 ± .00** | **.99 ± .03** | .82 ± .21 | **0.68 ± 0.07** | 6.60 ± 5.07 |

based methods for many cases. As discussed in Section 5.1, this is because the proposed method can detect the complex distribution difference more effectively than the Gaussian-based methods. Note that the proposed method also outperformed SPARDA. We conjecture that this was because SPARDA tended to be trapped by local optima when solving the nonconvex optimization.

The second finding is the computational efficacy of the proposed method. The proposed method was from 3 to more than 70 times faster than the entire runtime of SPARDA (see also Appendix 9).

The third finding is on the left two columns. The results show that the proposed method with the greedy scoring method attained comparable results with the exact solution of the sparsest $k$-subgraph problem (3). In other words, the greedy scoring method (Algorithm 1) provided good approximate solutions and can be a practical alternative for the exact method that may require exponential time. Note that the greedy scoring method sometimes outperformed the exact method (see Appendix 9 for the discussion).

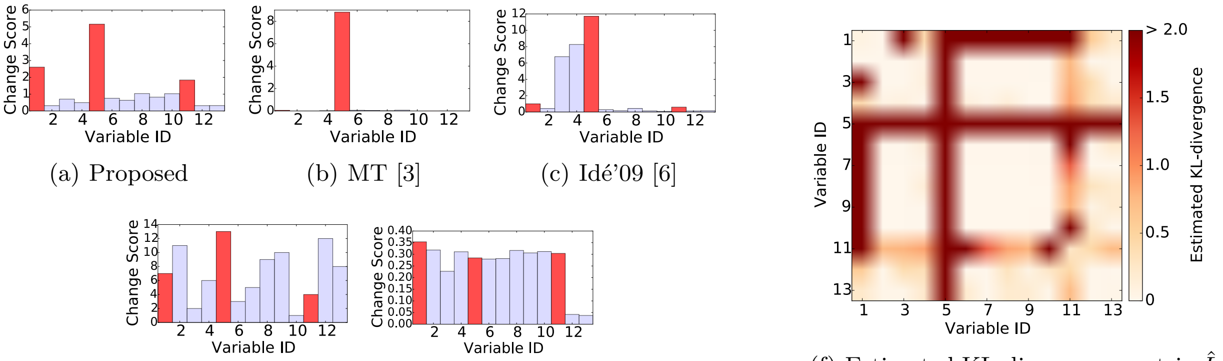
To demonstrate the success of the proposed method in detail, we display a result from the CBM dataset with Covariance Change ($c = 0.3$) in Figure 3. In this example, we set the features with distribution differences as $S = \{1, 5, 11\}$. In Figure 3(a), we observe that the score of the proposed method marked the top-three values on the set $S$, which is an ideal result. This is not the case with the other four baseline methods. The three Gaussian-based methods marked the largest score on the fifth feature, but they failed to detect the other two features. SPARDA marked the largest score on the first feature, but it failed to detect the other two features. The estimated KL-divergence matrix $\hat{L}$ in Figure 3(f) shows why the proposed method could detect differences successfully. $\hat{L}$ had large values on the rows and columns that corresponds to the set $S$. This means that Conditions (S1) and (S2) in Theorem 2 are met; thus, the set $S$ was detected properly.

### 5.3 Application to Anomaly Detection in Quantum Systems

We applied the proposed method to anomaly detection in quantum systems [17]. In quantum informatics, we sometimes face unknown errors in the given quantum state. For such cases, it is critically important to find the error sources for several applications, such as quantum computation, quantum cryptography, and quantum metrology.

In this experiment, we used data derived from a real physical experiment. In the physical experiment, 300 healthy density matrices were derived, each of which is a $4 \times 4$ Hermitian matrix. 50 anomalous matrices were also derived with a decoherence in their $(1, 4)$-th entry. See Appendix 10 for the details of the experimental settings. Experimentally obtained density matrices have the changes in both on the mean and variance on the $(1, 4)$-th entry (Table 4). Here, the task is to find the erroneous $(1, 4)$-th entry using different-feature selection. In the experiment, we applied two preprocessing. First, because the error appears only on the absolute value of the matrix entry, we computed the absolute value of each entry. Second, because the matrix is symmetric, we extracted only upper-triangular entries and transformed the matrix to a ten-dimensional vector.

Figure 4(a) shows that the proposed method and SPARDA attained AUROC=1 for all the decoherence levels (two lines overlapped in the figure). To exam-

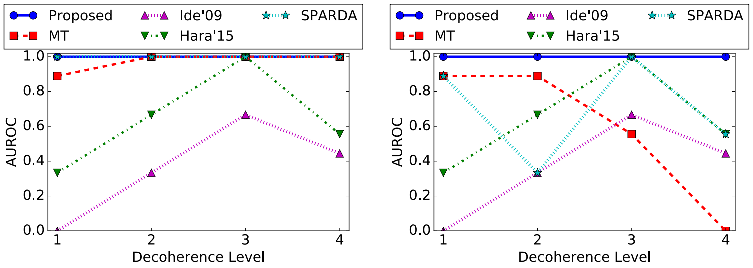


(a) Proposed (b) MT [3] (c) Idé'09 [6]

(d) Hara'15 [4] (e) SPARDA [5]

(f) Estimated KL-divergence matrix $\hat{L}$

Figure 3: Results on the CBM dataset with Covariance Change ($c = 0.3$): (a)–(e) Change Score $\hat{\boldsymbol{s}}$: (red bars on the 1st, 5th, and 11th features denote that they are features with distribution differences, while blue bars on the other features denote that they have no distribution differences); (f) Estimated KL-divergence matrix $\hat{L}$.



(a) Mean and Variance Change (b) Only Variance Change

Figure 4: AUROCs on decoherence data

Table 4: Decoherence level and the mean and the variance of $(1, 4)$-th entry. Level 0 is a healthy data.

| Level | Mean | Var. |
|---|---|---|
| 0 | 0.42 | 0.000651 |
| 1 | 0.40 | 0.001055 |
| 2 | 0.38 | 0.000876 |
| 3 | 0.36 | 0.000817 |
| 4 | 0.34 | 0.000768 |

ine the performance of each method in a more complex situation, we removed the mean change from the anomalous data. Here, the task is to identify the $(1, 4)$-th entry only from its variance changes. Figure 4(b) shows that, in this setting, only the proposed method attained AUROC$= 1$ for all decoherence levels. This result confirms that the proposed method could find different-features most effectively.

## 6 Conclusion

We have proposed a simple nonparametric method for different-feature selection that satisfies the two requirements, namely, less restrictive assumptions and computational efficiency. In the proposed method, we have first computed the KL-divergence matrix and then solved the sparsest $k$-subgraph problem derived from the matrix using a greedy scoring method. We have shown that the proposed method runs in only $O(D^2(N + M) \log NM)$ average time. Moreover, it does not require extra computation for model selection. We have also proved that the proposed method provides a consistent solution under mild conditions. In particular, it requires less restrictive assumptions on the data distributions for consistent estimation compared to current Gaussian-based methods.

The experimental results revealed that the proposed method significantly outperformed the Gaussian-based methods. The proposed method detected the complex distribution difference effectively and attained a high AUROC even for cases in which the Gaussian-based methods worked poorly. We also compared the proposed method to the state-of-the-art method SPARDA. The experimental results showed that the proposed method attained a higher AUROC than SPARDA on several datasets while requiring small computation time.

Despite the computational efficiency of the proposed method, there still remains a scalability issue. That is, the time complexity is proportional to $D^2$, which can be prohibitive in a high dimensional setting. Improving the computational scalability is one of our future directions.

## Acknowledgments

T. Katsuki was supported by CREST, JST. S. Takeuchi was supported by JSPS KAKENHI #26220712 and CREST, JST.

## References

[1] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995.

[2] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012.

[3] G. Taguchi and J. Rajesh. New trends in multivariate diagnosis. *The Indian Journal of Statistics, Series B*, pages 233–248, 2000.

[4] S. Hara, T. Morimura, T. Takahashi, H. Yanagisawa, and T. Suzuki. A consistent method for graph based anomaly localization. *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pages 333–341, 2015.

[5] J. W. Mueller and T. Jaakkola. Principal differences analysis: Interpretable characterization of differences between distributions. *Advances in Neural Information Processing Systems*, pages 1693–1701, 2015.

[6] T. Idé, A. C. Lozano, N. Abe, and Y. Liu. Proximity-based anomaly detection using sparse structure learning. *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 97–108, 2009.

[7] S. Hirose, K. Yamanishi, T. Nakata, and R. Fujimaki. Network anomaly detection based on eigen equation compression. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1185–1194, 2009.

[8] R. Jiang, H. Fei, and J. Huan. Anomaly localization for network data streams with graph joint sparse PCA. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 886–894, 2011.

[9] T. Idé, S. Papadimitriou, and M. Vlachos. Computing correlation anomaly scores using stochastic nearest neighbors. *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 523–528, 2007.

[10] A. L. Gibbs and F. E. Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3):419–435, 2002.

[11] R. Watrigant, M. Bougeret, and R. Giroudeau. Approximating the sparsest k-subgraph in chordal graphs. *Theory of Computing Systems*, 58(1):111–132, 2016.

[12] S. I. Lee, H. Lee, P. Abbeel, and A. Ng. Efficient l1 regularized logistic regression. *Proceedings of the National Conference on Artificial Intelligence*, 21(1):401, 2006.

[13] Q. Wang, S. R. Kulkarni, and S. Verdú. Divergence estimation for multidimensional densities via-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405, 2009.

[14] R. A. Brown. Building a balanced $k$-d tree in $\mathrm{O}(kn \log n)$ time. *Journal of Computer Graphics Techniques*, 4(1):50–68, 2015.

[15] A. Coraddu, L. Oneto, A. Ghio, S. Savio, D. Anguita, and M. Figari. Machine learning approaches for improving condition-based maintenance of naval propulsion plants. *Journal of Engineering for the Maritime Environment*, 2014.

[16] M. Lichman. UCI machine learning repository, 2013.

[17] S. Hara, T. Ono, R. Okamoto, T. Wahio, and S. Takeuchi. Anomaly detection in reconstructed quantum states using a machine-learning technique. *Physical Review A*, 89(2):022104, 2014.