

Classification with Nearest Disjoint Centroids

Nicolas Fraiman
fraiman@email.unc.edu

Zichao Li
lizichao@live.unc.edu

Abstract

In this paper, we develop a new classification method based on nearest centroid, and it is called the nearest disjoint centroid classifier. Our method differs from the nearest centroid classifier in the following two aspects: (1) the centroids are defined based on disjoint subsets of features instead of all the features, and (2) the distance is induced by the dimensionality-normalized norm instead of the Euclidean norm. We provide a few theoretical results regarding our method. In addition, we propose a simple algorithm based on adapted k -means clustering that can find the disjoint subsets of features used in our method, and extend the algorithm to perform feature selection. We evaluate and compare the performance of our method to other classification methods on both simulated data and real-world gene expression datasets. The results demonstrate that our method is able to outperform other competing classifiers by having smaller misclassification rates and/or using fewer features in various settings and situations.

1 Introduction

In general, classification is the task of predicting the category that an observation belongs to. Many applications in real life are dealing with classification problems, such as determining the category of images [Russakovsky et al., 2015], deciding the topic of documents [Lewis et al., 2004], and diagnosing the cancer type of tissues [Edgar et al., 2002]. In order to perform classification, people often build models that can automatically make predictions by identifying patterns in a dataset that includes observations and their associated class labels. Over the years, researchers have developed many different classification methods, ranging from simpler ones such as logistic regression [Hastie et al., 2009], k -nearest neighbors [Cover and Hart, 1967], Naive Bayes [Hand and Yu, 2001], and decision trees [Breiman et al., 1984], to more complicated ones such as support vector machines [Cortes and Vapnik, 1995], random forest [Breiman, 2001], gradient boosting machine [Friedman, 2001], and neural networks [Krizhevsky et al., 2012].

Among numerous existing classification methods, the nearest centroid classifier is one of the simplest classification method. It computes the centroid of each class as the average of the training samples that belong to that class, and classifies a test sample to the class with the nearest centroid. Intuitively, the nearest centroid classifier can be understood as creating one prototype to represent each class, and making predictions by selecting the class with the prototype that is most similar to the test sample. The theoretical simplicity makes it easy to understand and interpret, and the computational efficiency makes it appealing in practice. Therefore, it has been used in many different fields of application, including gene expression analysis [Tibshirani et al., 2002, Levner, 2005, Dabney, 2005, Dabney and Storey, 2007] and text classification [Han and Karypis, 2000, Lertnattee and Theeramunkong, 2004, Tan, 2008].

The nearest shrunken centroid classifier, a simple modification of the nearest centroid classifier, was proposed by Tibshirani et al. [2002]. It works by shrinking the class centroids toward the overall centroids after standardizing each feature by the pooled within-class standard deviation of that feature. Importantly, the shrinkage process can be considered as performing feature selection, which is desirable in applications with high-dimensional features. For example, when predicting the cancer type using a gene expression dataset, the nearest shrunken centroid classifier would select a small subset of the genes to make predictions, whereas the nearest centroid classifier would use all the genes. This characteristic makes the nearest shrunken centroid classifier a popular method in gene expression analysis [Tibshirani et al., 2003, Sørbye et al., 2003, Volinia et al., 2006, Parker et al., 2009, Curtis et al., 2012].

In this paper, we develop a new classification method based on nearest centroid, and it is called the nearest disjoint centroid classifier. The main idea is to define the centroids based on disjoint subsets of

features instead of all the features. More specifically, we partition the features into k groups, each one corresponding to one of the k classes, and the centroid for each class is defined using the corresponding group of features. In order to find the k disjoint subsets of features, we propose a simple algorithm based on adapted k -means clustering. A similar formulation was proposed in [Fraiman and Li \[2020\]](#), in which the authors used an alternating k -means algorithm with disjoint centroids to perform biclustering. However, our method applies to supervised classification problems rather than unsupervised biclustering problems. Importantly, this means that we assume there are k disjoint subsets of features with discriminative power, which is generally true for high-dimensional data where the number of features p is much larger than the number of classes k . In addition, our method is able to perform feature selection by adding a special cluster that represents a “global” baseline, and features assigned to the special cluster are not used in making predictions.

The rest of this paper is organized as follows. In [Section 2](#), we formulate the problem and give a high-level description of our nearest disjoint centroid classifier. In [Section 3](#), we present and prove a few theoretical results regarding our method. In [Section 4](#), we provide a rigorous proof of the main consistency result. In [Section 5](#), we propose a simple algorithm based on adapted k -means clustering that finds the disjoint subsets of features used in our method. In [Section 6](#), we extend our method to perform feature selection by assigning features to a special cluster that is not used for classification. In [Section 7](#), we evaluate and compare the performance of our nearest disjoint centroid classifier on simulated data to other classification methods. In [Section 8](#), we apply our method to three cancer gene expression datasets, and show that our method is able to outperform other competing classifiers by having smaller misclassification rates and/or using fewer features. In [Section 9](#), we conclude with a discussion.

2 Problem Formulation

Suppose we are given n training samples and their associated class labels $(X_1, Y_1), \dots, (X_n, Y_n)$ where $X_i \in \mathbb{R}^p$ and $Y_i \in \{1, \dots, k\}$ for $1 \leq i \leq n$. For $1 \leq j \leq k$, let S_j denote the set of indices of training samples that belong to class j . Throughout this paper, we assume $k \leq \min(n, p)$, and all S_j are non-empty. The nearest centroid classifier works by first computing the per-class centroid c_j as

$$c_j = \frac{1}{|S_j|} \sum_{i \in S_j} X_i, \quad 1 \leq j \leq k.$$

Then, it classifies a test sample X to the class Y with the nearest centroid. When using Euclidean distance, we have

$$Y = \arg \min_{1 \leq j \leq k} \|X - c_j\|_2^2.$$

Essentially, the centroids c_j are all vectors in \mathbb{R}^p , and they are computed by minimizing the following objective function:

$$\sum_{j=1}^k \sum_{i \in S_j} \|X_i - c_j\|_2^2.$$

Now, suppose the centroids are defined using disjoint subsets of features instead of all the features. More specifically, let $I = \{1, \dots, p\}$ be the index set of features, then I could be partitioned into k disjoint nonempty sets I_1, \dots, I_k , where $I_1 \cup \dots \cup I_k = I$. For any $X = (x_1, \dots, x_p) \in \mathbb{R}^p$, let $X(I_j) = (x_i)_{i \in I_j}$. The space of $X(I_j)$ is defined as \mathbb{R}^{I_j} , and we define the dimensionality-normalized norm on \mathbb{R}^{I_j} as

$$\|X(I_j)\|_{dn} = \sqrt{\frac{\sum_{i \in I_j} x_i^2}{l_j}},$$

where $l_j = |I_j|$ denote the cardinality of the index set I_j , and it is also the dimension of the space \mathbb{R}^{I_j} .

In our method, we would like to find the k disjoint subsets of features I_1, \dots, I_k and the corresponding k disjoint centroids $c_j \in \mathbb{R}^{I_j}, 1 \leq j \leq k$ such that the following objective function is minimized:

$$\sum_{j=1}^k \sum_{i \in S_j} \|X_i(I_j) - c_j\|_{dn}^2. \tag{1}$$

For a test sample X , we would classify it to the class Y with the nearest disjoint centroid (distance induced by the dimensionality-normalized norm), which is given by

$$Y = \arg \min_{1 \leq j \leq k} \|X(I_j) - c_j\|_{dn}^2.$$

The reason of using the dimensionality-normalized norm instead of the Euclidean norm in the objective function (1) is twofold:

1. Theoretically, it is important for each individual term $\|X(I_j) - c_j\|_{dn}^2$ to be appropriately normalized, so that the objective function (1) is summing up n roughly comparable terms no matter how large or small each subset of features I_j is. If we use the Euclidean norm in the objective function, then when the data is imbalanced (some classes have much more observations than other classes), the majority classes would be assigned much smaller subsets of features, and the minority classes would be assigned much larger subsets of features. This is because such assignment would minimize the objective function by minimizing the inner sum $\sum_{i \in S_j} \|X_i(I_j) - c_j\|_2^2$ for each class j .
2. Empirically, we conducted several simulations to compare the performance of our method using the dimensionality-normalized norm and the Euclidean norm. Although they are not included in the paper, the results confirmed our expectation that using Euclidean norm would lead to worse performance when the data is imbalanced.

It is easy to see that if the k disjoint subsets of features I_1, \dots, I_k are given, then the corresponding k centroids can be computed as

$$c_j = \frac{1}{|S_j|} \sum_{i \in S_j} X_i(I_j), \quad 1 \leq j \leq k.$$

However, finding the best disjoint subsets of features I_1, \dots, I_k to minimize the objective function (1) is a combinatorial optimization problem, which is computationally intractable for large p . In light of this fact, we present a simple algorithm in Section 5 that uses adapted k -means clustering to find the disjoint subsets of features I_1, \dots, I_k .

Again, we emphasize that our method assumes that there are k disjoint subsets of features with discriminative power, which might not apply to all data, but it is generally true for high-dimensional data where the number of features p is much larger than the number of classes k . In addition, it is possible to extend our method to handle the more general case by allowing the centroids to have a common set of features, which could be selected by running any feature selection algorithm. In that case, our main theoretical result Theorem 1 would still hold, and our main algorithms Algorithm 1 and Algorithm 2 would only need to be slightly modified.

3 Theoretical Results

Suppose the training samples and their associated class labels $(X_1, Y_1), \dots, (X_n, Y_n)$ are i.i.d. with the same distribution as (X, Y) where $X \in \mathbb{R}^p$ and $Y \in \{1, \dots, k\}$. Let μ denote the distribution of (X, Y) , and let μ_n denote the empirical distribution of the n training samples and their associated class labels. In addition, suppose that for $1 \leq j \leq k$, the probability of $Y = j$ is given by $p^{(j)}$:

$$P(Y = j) = p^{(j)},$$

and the conditional distribution of X given $Y = j$ is given by $\mu^{(j)}$:

$$X|Y = j \sim \mu^{(j)}.$$

Similarly, for $1 \leq j \leq k$, let $p_n^{(j)}$ denote the empirical proportion of $Y_i = j$, and let $\mu_n^{(j)}$ denote the empirical conditional distribution of X_i given $Y_i = j$ in the training data.

We minimize the empirical risk defined as

$$\begin{aligned} W(\mathbf{I}, \mathbf{c}, \mu_n) &= \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k \mathbf{1}_{\{Y_i=j\}} \cdot \|X_i(I_j) - c_j\|_{dn}^2 \\ &= \sum_{j=1}^k p_n^{(j)} \int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) \end{aligned}$$

over all feature subsets $\mathbf{I} = \{I_1, \dots, I_k\}$ and centroids $\mathbf{c} = \{c_1, \dots, c_k\}$. The risk is defined as

$$\begin{aligned} W(\mathbf{I}, \mathbf{c}, \mu) &= \int \sum_{j=1}^k \mathbf{1}_{\{y=j\}} \cdot \|x(I_j) - c_j\|_{dn}^2 d\mu(x, y) \\ &= \sum_{j=1}^k p^{(j)} \int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x). \end{aligned}$$

The optimal risk is defined as

$$W^*(\mu) = \inf_{\mathbf{I}} \inf_{\mathbf{c}} W(\mathbf{I}, \mathbf{c}, \mu).$$

For a fixed feature subset I_j , it is easy to verify that

$$\arg \min_{c_j} \int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) = \int x(I_j) d\mu_n^{(j)}(x),$$

and

$$\arg \min_{c_j} \int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x) = \int x(I_j) d\mu^{(j)}(x).$$

Let $\delta_n \geq 0$. A feature subsets \mathbf{I}_n and centroids \mathbf{c}_n as a whole is a δ_n -minimizer of the empirical risk if

$$W(\mathbf{I}_n, \mathbf{c}_n, \mu_n) \leq W^*(\mu_n) + \delta_n,$$

where $W^*(\mu_n) = \inf_{\mathbf{I}} \inf_{\mathbf{c}} W(\mathbf{I}, \mathbf{c}, \mu_n)$. When $\delta_n = 0$, \mathbf{I}_n and \mathbf{c}_n as a whole is called an empirical risk minimizer. Since μ_n is supported on at most n points, the existence of an empirical risk minimizer is guaranteed.

The first theoretical result of this paper is the following consistency theorem, which states that the risk of a δ_n -minimizer of the empirical risk converges to the optimal risk as long as $\lim_{n \rightarrow \infty} \delta_n = 0$.

Theorem 1. *Assume that all $\mu^{(j)}$ have finite second moments that are bounded by a constant h :*

$$\max_{1 \leq j \leq k} \int \|x\|_2^2 d\mu^{(j)}(x) \leq h.$$

Let \mathbf{I}_n and \mathbf{c}_n be a δ_n -minimizer of the empirical risk. If $\lim_{n \rightarrow \infty} \delta_n = 0$, then

$$\lim_{n \rightarrow \infty} W(\mathbf{I}_n, \mathbf{c}_n, \mu) = W^*(\mu) \text{ a.s.}$$

A detailed proof of Theorem 1 is given in Section 4, which also includes a remark at the end that provides some analysis on the rate of convergence.

We can characterize the feature subsets $\mathbf{I}^* = \{I_1^*, \dots, I_k^*\}$ and the centroids $\mathbf{c}^* = \{c_1^*, \dots, c_k^*\}$ that achieve the optimal risk $W^*(\mu)$ if we make some additional assumptions.

Theorem 2. *Assume that the feature subsets $\mathbf{I}^* = \{I_1^*, \dots, I_k^*\}$ is a partition of the p features $I = \{1, \dots, p\}$, and for $1 \leq j \leq k$ we have*

$$\min_{I_j \subset I} \int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x) = \int \|x(I_j^*) - c_j^*\|_{dn}^2 d\mu^{(j)}(x),$$

where $c_j = \int x(I_j) d\mu^{(j)}(x)$ and $c_j^ = \int x(I_j^*) d\mu^{(j)}(x)$. Then*

$$W(\mathbf{I}^*, \mathbf{c}^*, \mu) = W^*(\mu).$$

Proof. We prove by contradiction. Assume that there exist feature subsets $\mathbf{I}^\dagger = \{I_1^\dagger, \dots, I_k^\dagger\}$ and centroids $\mathbf{c}^\dagger = \{c_1^\dagger, \dots, c_k^\dagger\}$ such that

$$W(\mathbf{I}^\dagger, \mathbf{c}^\dagger, \mu) < W(\mathbf{I}^*, \mathbf{c}^*, \mu).$$

Since by definition

$$W(\mathbf{I}, \mathbf{c}, \mu) = \sum_{j=1}^k p^{(j)} \int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x),$$

there must exist at least one $t \in \{1, \dots, k\}$ such that

$$\int \|x(I_t^\dagger) - c_t^\dagger\|_{dn}^2 d\mu^{(t)}(x) < \int \|x(I_t^*) - c_t^*\|_{dn}^2 d\mu^{(t)}(x).$$

However, the property of \mathbf{I}^* guarantees that

$$\begin{aligned} \int \|x(I_t^*) - c_t^*\|_{dn}^2 d\mu^{(t)}(x) &= \min_{I_t \subset I} \int \|x(I_t) - c_t\|_{dn}^2 d\mu^{(t)}(x) \\ &\leq \int \|x(I_t^\dagger) - \int x(I_t^\dagger) d\mu^{(t)}(x)\|_{dn}^2 d\mu^{(t)}(x) \\ &\leq \int \|x(I_t^\dagger) - c_t^\dagger\|_{dn}^2 d\mu^{(t)}(x). \end{aligned}$$

where $c_t^* = \int x(I_t^*) d\mu^{(t)}(x)$ and $c_t = \int x(I_t) d\mu^{(t)}(x)$. Now we have a contradiction, and therefore such \mathbf{I}^\dagger and \mathbf{c}^\dagger could not exist. \square

Corollary 3. *Assume that the p features are all independent and consist of k successive blocks, each of size d . In addition, assume that for $1 \leq j \leq k$ and $X = (x_1, \dots, x_p) \sim \mu^{(j)}$, the d entries in the j -th block $x_{(j-1)d+1}, \dots, x_{jd}$ are i.i.d. with variance σ_1^2 , and the rest of the $p-d$ entries are i.i.d. with variance σ_2^2 , with $\sigma_1^2 < \sigma_2^2$. For $1 \leq j \leq k$, if we let*

$$I_j^* = \{x_{(j-1)d+1}, \dots, x_{jd}\},$$

and $c_j^* = \int x(I_j^*) d\mu^{(j)}(x)$, then $W(\mathbf{I}^*, \mathbf{c}^*, \mu) = W^*(\mu)$.

Proof. This is a direct corollary of Theorem 2. We only need to verify that for $1 \leq j \leq k$ we have

$$\min_{I_j \subset I} \int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x) = \int \|x(I_j^*) - c_j^*\|_{dn}^2 d\mu^{(j)}(x),$$

where $c_j = \int x(I_j) d\mu^{(j)}(x)$ and $c_j^* = \int x(I_j^*) d\mu^{(j)}(x)$. By assumption, for any specific feature $I_j = \{i\}$,

$$\int \|x(I_j) - c_j\|_2^2 d\mu^{(j)}(x) = \sigma_1^2$$

if $i \in I_j^*$, and

$$\int \|x(I_j) - c_j\|_2^2 d\mu^{(j)}(x) = \sigma_2^2 > \sigma_1^2$$

if $i \notin I_j^*$. This means that for any feature subset $I_j \subset I_j^*$, we have

$$\int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x) = \sigma_1^2.$$

In addition, for any other feature subset I_j that includes at least one feature $i \notin I_j^*$, we have

$$\int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x) > \sigma_1^2.$$

Hence for $1 \leq j \leq k$ we have

$$\min_{I_j \subset I} \int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x) = \sigma_1^2 = \int \|x(I_j^*) - c_j^*\|_{dn}^2 d\mu^{(j)}(x),$$

and the proof is completed. \square

4 Proof of the Main Theoretical Result

In this section, we give a detailed proof of Theorem 1, which is our main theoretical result. Recall that the L_2 Wasserstein distance between two probability measures μ_1 and μ_2 on \mathbb{R}^p , with finite second moment, is defined as

$$\gamma(\mu_1, \mu_2) = \inf_{X_1 \sim \mu_1, X_2 \sim \mu_2} (\mathbb{E} \|X_1 - X_2\|_2^2)^{1/2},$$

where the infimum is taken over all joint distributions of two random variables X_1 and X_2 such that X_1 has distribution μ_1 and X_2 has distribution μ_2 . It has been shown in [Rachev and Rüschendorf \[1998\]](#) that γ is a metric on the space of probability distributions on \mathbb{R}^p with finite second moment, and that the infimum is a minimum and can be achieved.

We first prove the following four lemmas.

Lemma 4. *For any feature subset I_j and centroid c_j , we have*

$$\left| \left[\int \|x(I_j) - c_j\|_{dn}^2 d\mu_1(x) \right]^{1/2} - \left[\int \|x(I_j) - c_j\|_{dn}^2 d\mu_2(x) \right]^{1/2} \right| \leq \gamma(\mu_1, \mu_2).$$

Proof. Let $X_1 \sim \mu_1$ and $X_2 \sim \mu_2$ achieve the infimum defining $\gamma(\mu_1, \mu_2)$. Then

$$\left[\int \|x(I_j) - c_j\|_{dn}^2 d\mu_1(x) \right]^{1/2} = \left[\mathbb{E} \frac{\|X_1(I_j) - c_j\|_2^2}{l_j} \right]^{1/2} \leq \left[\mathbb{E} \frac{(\|X_1(I_j) - X_2(I_j)\|_2 + \|X_2(I_j) - c_j\|_2)^2}{l_j} \right]^{1/2}.$$

Using Cauchy–Schwarz inequality, we have

$$\begin{aligned} & \mathbb{E} \left[\frac{(\|X_1(I_j) - X_2(I_j)\|_2 + \|X_2(I_j) - c_j\|_2)^2}{l_j} \right] \\ & \leq \mathbb{E} \|X_1 - X_2\|_2^2 + \mathbb{E} \left[\frac{\|X_2(I_j) - c_j\|_2^2}{l_j} \right] + 2\mathbb{E} \left[\|X_1 - X_2\|_2 \cdot \frac{\|X_2(I_j) - c_j\|_2}{\sqrt{l_j}} \right] \\ & \leq \mathbb{E} \|X_1 - X_2\|_2^2 + \mathbb{E} \left[\frac{\|X_2(I_j) - c_j\|_2^2}{l_j} \right] + 2 \left[\mathbb{E} \|X_1 - X_2\|_2^2 \right]^{1/2} \left[\mathbb{E} \frac{\|X_2(I_j) - c_j\|_2^2}{l_j} \right]^{1/2} \\ & = \left(\left[\mathbb{E} \|X_1 - X_2\|_2^2 \right]^{1/2} + \left[\mathbb{E} \frac{\|X_2(I_j) - c_j\|_2^2}{l_j} \right]^{1/2} \right)^2. \end{aligned}$$

Consequently

$$\begin{aligned} \left[\int \|x(I_j) - c_j\|_{dn}^2 d\mu_1(x) \right]^{1/2} & \leq \left[\mathbb{E} \|X_1 - X_2\|_2^2 \right]^{1/2} + \left[\mathbb{E} \frac{\|X_2(I_j) - c_j\|_2^2}{l_j} \right]^{1/2} \\ & = \gamma(\mu_1, \mu_2) + \left[\int \|x(I_j) - c_j\|_{dn}^2 d\mu_2(x) \right]^{1/2}, \end{aligned}$$

which implies that

$$\left| \left[\int \|x(I_j) - c_j\|_{dn}^2 d\mu_1(x) \right]^{1/2} - \left[\int \|x(I_j) - c_j\|_{dn}^2 d\mu_2(x) \right]^{1/2} \right| \leq \gamma(\mu_1, \mu_2).$$

The other direction can be proved similarly. □

Lemma 5. *For any feature subset I_j and centroid c_j , if*

$$\max \left(\int \|x(I_j) - c_j\|_{dn}^2 d\mu_1(x), \int \|x(I_j) - c_j\|_{dn}^2 d\mu_2(x) \right) \leq h,$$

then

$$\left| \int \|x(I_j) - c_j\|_{dn}^2 d\mu_1(x) - \int \|x(I_j) - c_j\|_{dn}^2 d\mu_2(x) \right| \leq 2\sqrt{h}\gamma(\mu_1, \mu_2).$$

Proof. Let $a = [\int \|x(I_j) - c_j\|_{dn}^2 d\mu_1(x)]^{1/2}$ and $b = [\int \|x(I_j) - c_j\|_{dn}^2 d\mu_2(x)]^{1/2}$. Then

$$\left| \int \|x(I_j) - c_j\|_{dn}^2 d\mu_1(x) - \int \|x(I_j) - c_j\|_{dn}^2 d\mu_2(x) \right| = |a^2 - b^2| = |a + b||a - b| \leq 2\sqrt{h}\gamma(\mu_1, \mu_2),$$

where the last inequality follows from Lemma 4. \square

Lemma 6. Recall that μ denote the distribution of (X, Y) , and is associated with $p^{(j)}$ and $\mu^{(j)}$ for $1 \leq j \leq k$. In addition, μ_n denote the empirical distribution of $(X_1, Y_1), \dots, (X_n, Y_n)$, and is associated with $p_n^{(j)}$ and $\mu_n^{(j)}$ for $1 \leq j \leq k$. For any feature subsets \mathbf{I} and centroids \mathbf{c} , if

$$\max_{I_j \in \mathbf{I}, c_j \in \mathbf{c}} \left(\int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x), \int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) \right) \leq h,$$

then

$$|W(\mathbf{I}, \mathbf{c}, \mu) - W(\mathbf{I}, \mathbf{c}, \mu_n)| \leq 2k\sqrt{h} \max_{1 \leq j \leq k} \gamma(\mu^{(j)}, \mu_n^{(j)}) + kh \max_{1 \leq j \leq k} |p^{(j)} - p_n^{(j)}|.$$

Proof. By triangle inequality, we have

$$\begin{aligned} |W(\mathbf{I}, \mathbf{c}, \mu) - W(\mathbf{I}, \mathbf{c}, \mu_n)| &= \left| \sum_{j=1}^k p^{(j)} \int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x) - \sum_{j=1}^k p_n^{(j)} \int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) \right| \\ &\leq \left| \sum_{j=1}^k p^{(j)} \int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x) - \sum_{j=1}^k p^{(j)} \int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) \right| \quad (2) \\ &\quad + \left| \sum_{j=1}^k p^{(j)} \int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) - \sum_{j=1}^k p_n^{(j)} \int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) \right|. \quad (3) \end{aligned}$$

Using Lemma 5, we have the following bound for the first term:

$$\begin{aligned} (2) &= \sum_{j=1}^k p^{(j)} \left| \int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x) - \int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) \right| \\ &\leq \sum_{j=1}^k p^{(j)} 2\sqrt{h}\gamma(\mu^{(j)}, \mu_n^{(j)}) \leq \sum_{j=1}^k 2\sqrt{h}\gamma(\mu^{(j)}, \mu_n^{(j)}) \leq 2k\sqrt{h} \max_{1 \leq j \leq k} \gamma(\mu^{(j)}, \mu_n^{(j)}). \end{aligned}$$

We have the following bound for the second term:

$$(3) = \sum_{j=1}^k \int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) |p^{(j)} - p_n^{(j)}| \leq \sum_{j=1}^k h |p^{(j)} - p_n^{(j)}| \leq kh \max_{1 \leq j \leq k} |p^{(j)} - p_n^{(j)}|.$$

Lemma 6 follows directly from the above three inequalities. \square

Lemma 7. For $1 \leq j \leq k$, $\lim_{n \rightarrow \infty} |p^{(j)} - p_n^{(j)}| = 0$ a.s., and $\lim_{n \rightarrow \infty} \gamma(\mu^{(j)}, \mu_n^{(j)}) = 0$ a.s.

Proof. It is well known that the empirical measure μ_n converges to μ almost surely. This implies that for $1 \leq j \leq k$, we have $p_n^{(j)}$ converges to $p^{(j)}$ almost surely, and $\mu_n^{(j)}$ converges to $\mu^{(j)}$ almost surely. For each $1 \leq j \leq k$, since $p_n^{(j)}$ converges to $p^{(j)}$ almost surely, we know that

$$\lim_{n \rightarrow \infty} |p^{(j)} - p_n^{(j)}| = 0 \text{ a.s.}$$

Since $\mu_n^{(j)}$ converges to $\mu^{(j)}$ almost surely, by Skorokhod's representation theorem, there exist $Z_n \sim \mu_n^{(j)}$ and $Z \sim \mu^{(j)}$ jointly distributed such that $Z_n \rightarrow Z$ almost surely. By the triangle inequality, we have

$$2\|Z_n\|_2^2 + 2\|Z\|_2^2 - \|Z_n - Z\|_2^2 \geq \|Z_n\|_2^2 + \|Z\|_2^2 - 2\|Z_n\|_2\|Z\|_2 \geq 0.$$

Hence Fatou's lemma implies

$$\liminf_{n \rightarrow \infty} \mathbb{E} [2\|Z_n\|_2^2 + 2\|Z\|_2^2 - \|Z_n - Z\|_2^2] \geq \mathbb{E} \left[\liminf_{n \rightarrow \infty} (2\|Z_n\|_2^2 + 2\|Z\|_2^2 - \|Z_n - Z\|_2^2) \right] = 4\mathbb{E}\|Z\|_2^2.$$

Since $\lim_{n \rightarrow \infty} \mathbb{E}\|Z_n\|_2^2 = \mathbb{E}\|Z\|_2^2$, we must have $\lim_{n \rightarrow \infty} \mathbb{E}\|Z_n - Z\|_2^2 = 0$, which implies that

$$\lim_{n \rightarrow \infty} \gamma(\mu^{(j)}, \mu_n^{(j)}) = 0 \text{ a.s.} \quad \square$$

Having proved the above four lemmas, we are ready to prove Theorem 1, which states the following: Assume that all $\mu^{(j)}$ have finite second moments that are bounded by a constant h :

$$\max_{1 \leq j \leq k} \int \|x\|_2^2 d\mu^{(j)}(x) \leq h.$$

Let \mathbf{I}_n and \mathbf{c}_n be a δ_n -minimizer of the empirical risk. If $\lim_{n \rightarrow \infty} \delta_n = 0$, then

$$\lim_{n \rightarrow \infty} W(\mathbf{I}_n, \mathbf{c}_n, \mu) = W^*(\mu) \text{ a.s.}$$

Proof. Let $\varepsilon > 0$ be arbitrary, and let \mathbf{I}^* and \mathbf{c}^* be any element satisfying

$$\inf_{\mathbf{I}} \inf_{\mathbf{c}} W(\mathbf{I}, \mathbf{c}, \mu) \leq W(\mathbf{I}^*, \mathbf{c}^*, \mu) < \inf_{\mathbf{I}} \inf_{\mathbf{c}} W(\mathbf{I}, \mathbf{c}, \mu) + \varepsilon. \quad (4)$$

Then

$$\begin{aligned} W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W^*(\mu) &= W(\mathbf{I}_n, \mathbf{c}_n, \mu) - \inf_{\mathbf{I}} \inf_{\mathbf{c}} W(\mathbf{I}, \mathbf{c}, \mu) \\ &\leq W(\mathbf{I}_n, \mathbf{c}_n, \mu) - (W(\mathbf{I}^*, \mathbf{c}^*, \mu) - \varepsilon) \\ &= W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W(\mathbf{I}_n, \mathbf{c}_n, \mu_n) + W(\mathbf{I}_n, \mathbf{c}_n, \mu_n) - W(\mathbf{I}^*, \mathbf{c}^*, \mu) + \varepsilon \\ &\leq W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W(\mathbf{I}_n, \mathbf{c}_n, \mu_n) + (W(\mathbf{I}^*, \mathbf{c}^*, \mu_n) + \delta_n) - W(\mathbf{I}^*, \mathbf{c}^*, \mu) + \varepsilon \\ &\leq |W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W(\mathbf{I}_n, \mathbf{c}_n, \mu_n)| + |W(\mathbf{I}^*, \mathbf{c}^*, \mu_n) - W(\mathbf{I}^*, \mathbf{c}^*, \mu)| + \delta_n + \varepsilon. \end{aligned}$$

We now further analyze the right hand side of the last inequality:

$$W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W^*(\mu) \leq |W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W(\mathbf{I}_n, \mathbf{c}_n, \mu_n)| + |W(\mathbf{I}^*, \mathbf{c}^*, \mu_n) - W(\mathbf{I}^*, \mathbf{c}^*, \mu)| + \delta_n + \varepsilon. \quad (5)$$

For the first term $|W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W(\mathbf{I}_n, \mathbf{c}_n, \mu_n)|$, recall that \mathbf{I}_n and \mathbf{c}_n is a δ_n -minimizer of the empirical risk. This means that for each $I_n^{(j)} \in \mathbf{I}_n$, the corresponding $c_n^{(j)} \in \mathbf{c}_n$ is selected to minimize $\int \|x(I_n^{(j)}) - c_n^{(j)}\|_2^2 d\mu_n^{(j)}(x)$, and it can be written as $c_n^{(j)} = \int x(I_n^{(j)}) d\mu_n^{(j)}(x)$. Note that for each $I_n^{(j)} \in \mathbf{I}_n$ and the corresponding $c_n^{(j)} \in \mathbf{c}_n$, we have

$$\int \|x(I_n^{(j)}) - c_n^{(j)}\|_2^2 d\mu_n^{(j)}(x) \leq \int \|x(I_n^{(j)}) - c_n^{(j)}\|_2^2 d\mu_n^{(j)}(x) \leq \int \|x - b_n^{(j)}\|_2^2 d\mu_n^{(j)}(x),$$

where $b_n^{(j)} = \int x d\mu_n^{(j)}(x)$. Since $\mu_n^{(j)}$ converges to $\mu^{(j)}$ almost surely, by the strong law of large numbers, we know that

$$b_n^{(j)} = \int x d\mu_n^{(j)}(x) \xrightarrow{\text{a.s.}} \int x d\mu^{(j)}(x) = b_j,$$

and

$$\int \|x - b_n^{(j)}\|_2^2 d\mu_n^{(j)}(x) \xrightarrow{\text{a.s.}} \int \|x - b_j\|_2^2 d\mu^{(j)}(x).$$

Similarly, for each $I_n^{(j)} \in \mathbf{I}_n$ and the corresponding $c_n^{(j)} \in \mathbf{c}_n$, we have

$$\int \|x(I_n^{(j)}) - c_n^{(j)}\|_2^2 d\mu^{(j)}(x) \leq \int \|x(I_n^{(j)}) - c_n^{(j)}\|_2^2 d\mu^{(j)}(x) \leq \int \|x - b_n^{(j)}\|_2^2 d\mu^{(j)}(x),$$

and

$$\int \|x - b_n^{(j)}\|_2^2 d\mu^{(j)}(x) \xrightarrow{a.s.} \int \|x - b_j\|_2^2 d\mu^{(j)}(x).$$

Note that

$$\int \|x - b_j\|_2^2 d\mu^{(j)}(x) \leq \int \|x\|_2^2 d\mu^{(j)}(x) \leq h.$$

Therefore, for each j we can select N_j such that for all $n \geq N_j$, with probability 1 we have

$$\int \|x(I_n^{(j)}) - c_n^{(j)}\|_{dn}^2 d\mu_n^{(j)}(x) \leq \int \|x - b_n^{(j)}\|_2^2 d\mu_n^{(j)}(x) \leq 2h,$$

and

$$\int \|x(I_n^{(j)}) - c_n^{(j)}\|_{dn}^2 d\mu^{(j)}(x) \leq \int \|x - b_n^{(j)}\|_2^2 d\mu^{(j)}(x) \leq 2h.$$

Let $N_0 = \max_{1 \leq j \leq k} N_j$, then for all $n \geq N_0$, with probability 1 we have

$$\max_{I_n^{(j)} \in \mathbf{I}_n, c_n^{(j)} \in \mathbf{c}_n} \left(\int \|x(I_n^{(j)}) - c_n^{(j)}\|_{dn}^2 d\mu^{(j)}(x), \int \|x(I_n^{(j)}) - c_n^{(j)}\|_{dn}^2 d\mu_n^{(j)}(x) \right) \leq 2h.$$

Using Lemma 6, for all $n \geq N_0$, with probability 1 we have

$$|W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W(\mathbf{I}_n, \mathbf{c}_n, \mu_n)| \leq 2k\sqrt{2h} \max_{1 \leq j \leq k} \gamma(\mu^{(j)}, \mu_n^{(j)}) + 2kh \max_{1 \leq j \leq k} |p^{(j)} - p_n^{(j)}|. \quad (6)$$

For the second term $|W(\mathbf{I}^*, \mathbf{c}^*, \mu_n) - W(\mathbf{I}^*, \mathbf{c}^*, \mu)|$, for each $I_j \in \mathbf{I}^*$, we can write the corresponding $c_j \in \mathbf{c}^*$ as $c_j = \int x(I_j) d\mu^{(j)}(x)$ in order to minimize $W(\mathbf{I}^*, \mathbf{c}^*, \mu)$. Similar to the steps in bounding the first term, for each $I_j \in \mathbf{I}^*$ and the corresponding $c_j \in \mathbf{c}^*$, we have

$$\int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) \leq \int \|x(I_j) - c_j\|_2^2 d\mu_n^{(j)}(x) \leq \int \|x - b_j\|_2^2 d\mu_n^{(j)}(x) \xrightarrow{a.s.} \int \|x - b_j\|_2^2 d\mu^{(j)}(x),$$

and

$$\int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x) \leq \int \|x(I_j) - c_j\|_2^2 d\mu^{(j)}(x) \leq \int \|x - b_j\|_2^2 d\mu^{(j)}(x).$$

Therefore, for each j we can select M_j such that for all $n \geq M_j$, with probability 1 we have

$$\int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) \leq \int \|x - b_j\|_2^2 d\mu_n^{(j)}(x) \leq 2h,$$

and

$$\int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x) \leq \int \|x - b_j\|_2^2 d\mu^{(j)}(x) \leq 2h.$$

Let $M_0 = \max_{1 \leq j \leq k} M_j$, then for all $n \geq M_0$, with probability 1 we have

$$\max_{I_j \in \mathbf{I}^*, c_j \in \mathbf{c}^*} \left(\int \|x(I_j) - c_j\|_{dn}^2 d\mu^{(j)}(x), \int \|x(I_j) - c_j\|_{dn}^2 d\mu_n^{(j)}(x) \right) \leq 2h.$$

Using Lemma 6, for all $n \geq M_0$, with probability 1 we have

$$|W(\mathbf{I}^*, \mathbf{c}^*, \mu_n) - W(\mathbf{I}^*, \mathbf{c}^*, \mu)| \leq 2k\sqrt{2h} \max_{1 \leq j \leq k} \gamma(\mu^{(j)}, \mu_n^{(j)}) + 2kh \max_{1 \leq j \leq k} |p^{(j)} - p_n^{(j)}|. \quad (7)$$

Combining the inequalities (5), (6), (7), for all $n \geq \max(N_0, M_0)$, with probability 1 we have

$$W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W^*(\mu) \leq 4k\sqrt{2h} \max_{1 \leq j \leq k} \gamma(\mu^{(j)}, \mu_n^{(j)}) + 4kh \max_{1 \leq j \leq k} |p^{(j)} - p_n^{(j)}| + \delta_n + \varepsilon.$$

Using Lemma 7, we know that $\max_{1 \leq j \leq k} \gamma(\mu^{(j)}, \mu_n^{(j)}) \xrightarrow{a.s.} 0$, and $\max_{1 \leq j \leq k} |p^{(j)} - p_n^{(j)}| \xrightarrow{a.s.} 0$. Since ε is arbitrary and $\lim_{n \rightarrow \infty} \delta_n = 0$, we have

$$W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W^*(\mu) \xrightarrow{a.s.} 0. \quad \square$$

Remark. If we slightly modify inequality (4) and choose ε_n such that $\lim_{n \rightarrow \infty} \varepsilon_n = 0$, and let \mathbf{I}_n^* and \mathbf{c}_n^* be any element satisfying

$$\inf_{\mathbf{I}} \inf_{\mathbf{c}} W(\mathbf{I}, \mathbf{c}, \mu) \leq W(\mathbf{I}_n^*, \mathbf{c}_n^*, \mu) < \inf_{\mathbf{I}} \inf_{\mathbf{c}} W(\mathbf{I}, \mathbf{c}, \mu) + \varepsilon_n. \quad (8)$$

Then, similar to inequality (5), we could obtain

$$W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W^*(\mu) \leq |W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W(\mathbf{I}_n, \mathbf{c}_n, \mu_n)| + |W(\mathbf{I}_n^*, \mathbf{c}_n^*, \mu_n) - W(\mathbf{I}_n^*, \mathbf{c}_n^*, \mu)| + \delta_n + \varepsilon_n. \quad (9)$$

Also, similar to inequality (7), we could prove that for all $n \geq M_0$, with probability 1 we have

$$|W(\mathbf{I}_n^*, \mathbf{c}_n^*, \mu_n) - W(\mathbf{I}_n^*, \mathbf{c}_n^*, \mu)| \leq 2k\sqrt{2h} \max_{1 \leq j \leq k} \gamma(\mu^{(j)}, \mu_n^{(j)}) + 2kh \max_{1 \leq j \leq k} |p^{(j)} - p_n^{(j)}|. \quad (10)$$

Combining the inequalities (6), (9), (10), for all $n \geq \max(N_0, M_0)$, with probability 1 we have

$$W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W^*(\mu) \leq 4k\sqrt{2h} \max_{1 \leq j \leq k} \gamma(\mu^{(j)}, \mu_n^{(j)}) + 4kh \max_{1 \leq j \leq k} |p^{(j)} - p_n^{(j)}| + \delta_n + \varepsilon_n.$$

Now, if we assume there exist β_n such that for $1 \leq j \leq k$, we have

$$\lim_{n \rightarrow \infty} \beta_n |p^{(j)} - p_n^{(j)}| < \infty \text{ a.s.}, \text{ and } \lim_{n \rightarrow \infty} \beta_n \gamma(\mu^{(j)}, \mu_n^{(j)}) < \infty \text{ a.s.} \quad (11)$$

Then, as long as we choose δ_n and ε_n such that $\lim_{n \rightarrow \infty} \beta_n \delta_n < \infty$ and $\lim_{n \rightarrow \infty} \beta_n \varepsilon_n < \infty$, we have

$$\begin{aligned} & \lim_{n \rightarrow \infty} \beta_n [W(\mathbf{I}_n, \mathbf{c}_n, \mu) - W^*(\mu)] \\ & \leq 4k\sqrt{2h} \max_{1 \leq j \leq k} \lim_{n \rightarrow \infty} \beta_n \gamma(\mu^{(j)}, \mu_n^{(j)}) + 4kh \max_{1 \leq j \leq k} \lim_{n \rightarrow \infty} \beta_n |p^{(j)} - p_n^{(j)}| + \lim_{n \rightarrow \infty} \beta_n \delta_n + \lim_{n \rightarrow \infty} \beta_n \varepsilon_n \\ & \leq \infty \text{ a.s.} \end{aligned}$$

There are a few choices for β_n that satisfy inequality (11), depending on the assumptions on μ_j and p_j . The earliest result is by [Ajtai et al. \[1984\]](#) for the Lebesgue measure which was later sharpened by [Dobrić and Yukich \[1995\]](#) to $\beta_n = n^{1/p}$. Theorem 11.1.6 of [Rachev \[1991\]](#) generalizes [Dudley \[1969\]](#) to show that under a metric entropy condition, we could let $\beta_n = n^{2/p}$. [Horowitz and Karandikar \[1994\]](#) proved that under some very weak assumptions, we could let $\beta_n = n^{2/(p+4)}$. Some refinements were provided in [Fournier and Guillin \[2015\]](#) and [Weed and Bach \[2019\]](#).

5 Algorithm

In this section, we present a simple algorithm that outputs the k disjoint subsets of features I_1, \dots, I_k . The idea is to first transpose the $n \times p$ data matrix and then use an adapted version of the k -means clustering algorithm to produce a partition of the p rows I_1, \dots, I_k . The algorithm works as shown in Algorithm 1.

Recall that S_j denote the set of indices of training samples that belong to class j . After we have obtained the k disjoint subsets of features I_1, \dots, I_k , we can compute the k disjoint centroids that represent the k classes using the following equation:

$$c_j = \frac{1}{|S_j|} \sum_{i \in S_j} X_i(I_j), \quad 1 \leq j \leq k. \quad (12)$$

To classify a new data point, we simply choose the class with the nearest disjoint centroid (distance induced by the dimensionality-normalized norm).

In practice, it is recommended to run our algorithm multiple times to produce multiple partitions, and choose the partition that results in the lowest training error. There are mainly two reasons:

1. First, our algorithm depends on the partition obtained from the initial k -means clustering, which itself is not deterministic and might provide different results in different runs. Therefore, running our algorithm multiple times increases the probability of finding a partition that gives better performance.
2. Second, just like k -means clustering algorithm, our algorithm also might encounter empty cluster problem, although the probability is small when k is much smaller than p . More specifically, if in any assignment step any group of row indices $I_j, 1 \leq j \leq k$ becomes empty, then the algorithm cannot proceed and need to restart.

Algorithm 1 Finding the k disjoint subsets of features I_1, \dots, I_k (no feature selection)

1. Start by transposing the $n \times p$ data matrix and then performing k -means clustering on the rows to obtain k clusters as the initial partition of the p rows I_1, \dots, I_k .
2. We use an alternating procedure to update I_1, \dots, I_k . It is quite similar to the Lloyd’s algorithm in k -means clustering, except that the cluster centers m_j are defined on \mathbb{R}^{S_j} instead of \mathbb{R}^n , and the distance function is induced by the dimensionality-normalized norm instead of the Euclidean norm:
 - (a) (Update step) Given row partitions I_1, \dots, I_k , update the cluster centers m_1, \dots, m_k by

$$m_j = \frac{1}{|I_j|} \sum_{i \in I_j} T_i(S_j), 1 \leq j \leq k,$$

where T_i denotes the i -th row of the transposed $p \times n$ data matrix.

- (b) (Assignment step) Given cluster centers m_1, \dots, m_k , update the row partitions I_1, \dots, I_k by assigning every row to the cluster center with the smallest distance (induced by the dimensionality-normalized norm), and all the rows that are closest to m_j form $I_j, 1 \leq j \leq k$.

Alternate between (a) and (b) until convergence, and obtain a partition of the p rows I_1, \dots, I_k .

6 Feature Selection

In this section, we consider extending Algorithm 1 to perform feature selection. So far, although our method partitions features into disjoint subsets, it still uses all the features to classify new data points. However, in many applications such as gene expression analysis, the data usually include thousands of genes, many of which can be considered as irrelevant to predicting certain types of cancer. In those situations, it would be desirable if our method could perform feature selection, namely selecting a subset of the features so that only those features are used in making predictions.

In order to enable our method to perform feature selection, we need to make some small adjustments to Algorithm 1. More specifically, we add a new special cluster I_0 , and the features that belong to this special cluster are not used in prediction. Its cluster center m_0 are defined on \mathbb{R}^{S_0} , and we define $S_0 = \{1, \dots, n\}$. Intuitively, this cluster center represents a “global” baseline that is based on all data points, as opposed to other cluster centers m_j that represent class-specific baselines that are based on data points specific to class j . When classifying new data points, we still choose the class with the nearest disjoint centroid among c_1, \dots, c_k (distance induced by the dimensionality-normalized norm), which in turn depends on features in I_1, \dots, I_k . In this way, the features in I_0 are not used at all in making predictions, and therefore our method can be considered as performing feature selection.

The modified algorithm that is able to perform feature selection works as shown in Algorithm 2. As a way to control the number of selected features, we introduce a tuning parameter λ . When computing the distances (induced by the dimensionality-normalized norm) to the clusters centers in the assignment step, the distance to m_0 is multiplied by a factor λ . When $\lambda = \infty$, the distance to m_0 will become ∞ , so no feature will be assigned to the special cluster I_0 , which means that all features will be selected. As λ gets smaller and smaller, the distance to m_0 will become smaller and smaller, so features are more and more likely to be assigned to the special cluster I_0 , which means that less and less features are getting selected. In practice, λ can be considered as a hyperparameter that needs to be tuned based on the data, because it affects both the number of selected features and the performance of the model.

After we have obtained the $k + 1$ disjoint subsets of features I_0, \dots, I_k , the rest of the process is exactly the same as before: compute the k disjoint centroids c_1, \dots, c_k based on I_1, \dots, I_k using Equation (12), and classify a new data point to the class with the nearest disjoint centroid (distance induced by the dimensionality-normalized norm). Noticeably, the features in I_0 are not involved in the computation of the distances to the k disjoint centroids c_1, \dots, c_k , therefore they are not used in prediction.

Algorithm 2 Finding the $k + 1$ disjoint subsets of features I_0, \dots, I_k (with feature selection)

1. Start by transposing the $n \times p$ data matrix and then performing k -means clustering on the rows to obtain $k + 1$ clusters as the initial partition of the p rows I_0, \dots, I_k .
2. We use an alternating procedure to update I_0, \dots, I_k . It is quite similar to the Lloyd’s algorithm in k -means clustering, except that the cluster centers m_j are defined on \mathbb{R}^{S_j} instead of \mathbb{R}^n , and the distance function is induced by the dimensionality-normalized norm instead of the Euclidean norm:
 - (a) (Update step) Given row partitions I_0, \dots, I_k , update the cluster centers m_0, \dots, m_k by

$$m_j = \frac{1}{|I_j|} \sum_{i \in I_j} T_i(S_j), 0 \leq j \leq k,$$

where T_i denotes the i -th row of the transposed $p \times n$ data matrix.

- (b) (Assignment step) Given cluster centers m_0, \dots, m_k , update the row partitions I_0, \dots, I_k by assigning every row to the cluster center with the smallest distance (induced by the dimensionality-normalized norm), and all the rows that are closest to m_j form $I_j, 0 \leq j \leq k$. Note that the distance to m_0 is multiplied by a factor λ , which is a tuning parameter.

Alternate between (a) and (b) until convergence, and obtain a partition of the p rows I_0, \dots, I_k .

7 Simulation Studies

In this section, we evaluate and compare the performance of seven classification methods on simulated data with different settings. The first three classification methods are all based on nearest centroid:

1. Nearest disjoint centroid (NDC): This is the method presented in this paper. We consider both versions of our nearest disjoint centroid method, with and without feature selection. Algorithm 1 (NDC) is the version without feature selection, and Algorithm 2 (NDC-S) is the version with feature selection. For both algorithms, we run 100 times and pick the best partition, as suggested at the end of Section 5.
2. Nearest centroid (NC): This method simply classifies every data point to the class with the nearest centroid, and each centroid is defined as the average of the data points that belong to each class.
3. Nearest shrunken centroid (NSC) [Tibshirani et al., 2002]: This method is a simple modification of the nearest centroid method. It shrinks the class centroids toward the overall centroid after standardizing by the within-class standard deviation. The shrinkage process achieves feature selection.

In addition, we also include the following four widely used classification methods: k -nearest neighbors (KNN) [Cover and Hart, 1967], penalized linear discriminant analysis (LDA) [Witten and Tibshirani, 2011], support vector machine (SVM) [Cortes and Vapnik, 1995], and logistic regression with L_1 regularization (Logistic) [Hastie et al., 2009]. Note that SVM also has a sparse version [Yi and Huang, 2017] that could perform feature selection, but the implementation only works for binary classification, so it is not used in the simulations. The evaluation metric is the misclassification rate. The number of neighbors in KNN is set to be 15. Other hyperparameters, including the λ in NDC-S, LDA, and Logistic, and the threshold Δ in NSC, are chosen via cross-validation on the training set.

We generate simulated data in four different settings. In all settings, the data matrix satisfies the following property:

1. The rows have k blocks, each of size n . They represent k classes, each consisting of n data points.
2. The columns also have k blocks, each of size d . They represent k groups of features, each of size d .
3. As a whole, the data matrix consists of $k \times k$ blocks, each of size $n \times d$.

4. All the entries in the data matrix are independent. In addition, the entries in each small $n \times d$ block are identically distributed. The entries in the k blocks on the main diagonal of the data matrix follow $\mathcal{N}(\mu_1, \sigma_1^2)$, and all the other entries in the data matrix follow $\mathcal{N}(\mu_2, \sigma_2^2)$.

In all settings, we set $k = 4$ and $n = 250$. The rest of the parameters, including $d, \mu_1, \mu_2, \sigma_1, \sigma_2$, might vary in different settings. For each setting, we perform 50 simulations, and report the means and standard errors of the misclassification rates. In each simulation, we generate two data matrices, one as training set and the other as test set.

7.1 Simulation 1: Blocks with Different Means and the Same Variance

In the first simulation, we consider the case where the $k \times k$ blocks have different means and the same variance. More specifically, we set $\mu_1 = a$, and $\mu_2 = 0$, where $a \in \{0.3, 0.6, 0.9\}$. As a increases, the difference between the means in different blocks also increases. In addition, we set $\sigma_1 = \sigma_2 = 1$, which means that all entries have the same standard deviation of 1. We also set $d \in \{3, 5, 10\}$. As d increases, the number of features in each block also increases.

	NDC	NDC-S	NC	NSC	KNN	LDA	SVM	Logistic
$d = 3$								
$a = 0.3$	0.741	0.742	0.614	0.616	0.679	0.617	0.637	0.612
$a = 0.6$	0.666	0.679	0.444	0.446	0.516	0.446	0.470	0.444
$a = 0.9$	0.549	0.585	0.286	0.287	0.331	0.287	0.304	0.286
$d = 5$								
$a = 0.3$	0.736	0.737	0.568	0.572	0.656	0.568	0.593	0.567
$a = 0.6$	0.622	0.654	0.353	0.355	0.436	0.354	0.373	0.352
$a = 0.9$	0.468	0.508	0.177	0.178	0.222	0.177	0.192	0.182
$d = 10$								
$a = 0.3$	0.724	0.731	0.489	0.490	0.607	0.491	0.505	0.490
$a = 0.6$	0.558	0.596	0.208	0.211	0.297	0.210	0.228	0.215
$a = 0.9$	0.342	0.468	0.059	0.060	0.089	0.060	0.069	0.067

Table 1: The means of the misclassification rate for Simulation 1 over 50 simulations. Most of the standard errors are less than 0.003, and the largest standard error is 0.011.

Results are reported in Table 1. In this setting, we see that NC, NSC, LDA, and Logistic have extremely similar and also the smallest misclassification rates, followed closely by SVM and KNN, and finally NDC and NDC-S with significantly larger misclassification rates. In addition, we observe a general pattern that as d and a increase, the misclassification rates decrease. This pattern makes intuitive sense, because larger d means more features, and larger a means larger difference between the means in different blocks, both of which should improve the performance of classification methods.

It is important to point out that our method (NDC and NDC-S) performing worse than other competing classification methods, including NC and NSC which are directly comparable, is expected in this setting. The reason is that all the features provide useful signals by having different means across different classes, and there is no benefit in considering features separately since they all have the same variance. Our method defines centroids using disjoint subsets of features, thereby losing valuable information compared to NC and NSC, both of which define centroids using all the features. However, when different features have different variances, our method would perform much better than all other classification methods, as we will see in the following simulations.

7.2 Simulation 2: Blocks with Different Variances and the Same Mean

In the second simulation, we consider the case where the $k \times k$ blocks have different variances and the same mean. More specifically, we set $\sigma_1 = 1$, and $\sigma_2 = 1 + b$, where $b \in \{0.3, 0.6, 0.9\}$. As b increases, the difference between the variances in different blocks also increases. In addition, we set $\mu_1 = \mu_2 = 0$, which means that

all entries have the same mean of 0. We also set $d \in \{3, 5, 10\}$. As d increases, the number of features in each block also increases.

	NDC	NDC-S	NC	NSC	KNN	LDA	SVM	Logistic
$d = 3$								
$b = 0.3$	0.696	0.694	0.745	0.746	0.708	0.747	0.677	0.750
$b = 0.6$	0.527	0.608	0.737	0.735	0.637	0.738	0.584	0.746
$b = 0.9$	0.360	0.403	0.735	0.731	0.575	0.733	0.503	0.746
$d = 5$								
$b = 0.3$	0.587	0.643	0.747	0.749	0.704	0.749	0.679	0.751
$b = 0.6$	0.359	0.379	0.744	0.743	0.628	0.743	0.557	0.746
$b = 0.9$	0.231	0.245	0.739	0.736	0.554	0.739	0.457	0.745
$d = 10$								
$b = 0.3$	0.430	0.449	0.746	0.747	0.703	0.744	0.691	0.751
$b = 0.6$	0.190	0.202	0.741	0.737	0.618	0.740	0.580	0.746
$b = 0.9$	0.073	0.079	0.737	0.731	0.546	0.736	0.485	0.745

Table 2: The means of the misclassification rate for Simulation 2 over 50 simulations. Most of the standard errors are less than 0.003, and the largest standard error is 0.008.

Results are reported in Table 2. In this setting, we see that NC, NSC, LDA, and Logistic also have extremely similar but the worst performance, with misclassification rates around 0.75 (equivalent to random guessing) in all cases. These numbers indicate that NC, NSC, LDA, and Logistic are all unable to detect heteroskedastic structure in the data, regardless of the value of d and b . SVM and KNN perform slightly better, although their misclassification rates are still much larger than those of NDC and NDC-S, both of which clearly outperform all other competing classification methods.

The reason that our method performs well in this setting lies in the fact that different features have different variances across different classes. Although different features have the same mean so the centroids have the same value, by defining centroids using disjoint subsets of features, different variances across different classes lead to different distances to different centroids. In addition, Corollary 3 guarantees that when $\sigma_1 < \sigma_2$, we can obtain the appropriate disjoint subsets of features.

7.3 Simulation 3: Blocks with Different Means and Different Variances

In the third simulation, we consider the case where the $k \times k$ blocks have different means and different variances, which is a combination of the first and second case. More specifically, we set $\mu_1 = c, \sigma_1 = 1$, and $\mu_2 = 0, \sigma_2 = 1 + c$, where $c \in \{0.3, 0.6, 0.9\}$. As c increases, the difference between the means and variances in different blocks also increases. We also set $d \in \{3, 5, 10\}$. As d increases, the number of features in each block also increases.

Results are reported in Table 3. In this setting, we see that in general, the ranking of different classification methods is similar to the ranking in Simulation 2: NDC and NDC-S clearly have the best performance, followed by SVM and KNN, and NC, NSC, LDA, and Logistic have similar but the worst performance. Comparing the results in Table 3 to those in Table 1, we notice that the additional difference between the variances in different blocks significantly helps NDC and NDC-S, leading to much smaller misclassification rates. In contrast, the misclassification rates of all other classification methods increase significantly after introducing the additional difference between block variances. Importantly, in this simulation, larger c means larger difference between both the means and the variances in different blocks, so both kinds of signals are present in the data. In this situation, NDC and NDC-S outperform all other competing classifiers, which indicates that our method could potentially obtain competitive performance when dealing with complex datasets in the real world.

7.4 Simulation 4: Adding Irrelevant Features

In the fourth simulation, we study the impact of adding irrelevant features on different classification methods. More specifically, we fix $d = 5$, and the first 20 columns of the data matrix is the same as the data matrix in

	NDC	NDC-S	NC	NSC	KNN	LDA	SVM	Logistic
$d = 3$								
$c = 0.3$	0.674	0.679	0.663	0.668	0.669	0.668	0.629	0.664
$c = 0.6$	0.442	0.514	0.583	0.585	0.547	0.583	0.499	0.582
$c = 0.9$	0.295	0.381	0.512	0.513	0.442	0.511	0.398	0.511
$d = 5$								
$c = 0.3$	0.551	0.620	0.631	0.632	0.650	0.635	0.593	0.631
$c = 0.6$	0.302	0.324	0.511	0.513	0.484	0.514	0.426	0.507
$c = 0.9$	0.165	0.182	0.413	0.414	0.357	0.413	0.315	0.409
$d = 10$								
$c = 0.3$	0.395	0.419	0.561	0.563	0.608	0.562	0.533	0.564
$c = 0.6$	0.136	0.144	0.381	0.383	0.395	0.381	0.328	0.379
$c = 0.9$	0.038	0.043	0.253	0.254	0.240	0.255	0.203	0.254

Table 3: The means of the misclassification rate for Simulation 3 over 50 simulations. Most of the standard errors are less than 0.003, and the largest standard error is 0.009.

Simulation 3, where we set $\mu_1 = c, \sigma_1 = 1, \mu_2 = 0, \sigma_2 = 1 + c$, and $c \in \{0.3, 0.6, 0.9\}$. However, the remaining r columns of the data matrix are r irrelevant features consisting of i.i.d. standard Gaussian variables, where $r \in \{20, 40, 80\}$.

	NDC	NDC-S	NC	NSC	KNN	LDA	SVM	Logistic
$r = 20$								
$c = 0.3$	0.550	0.570	0.635	0.638	0.668	0.640	0.627	0.634
$c = 0.6$	0.354	0.330	0.515	0.518	0.526	0.521	0.492	0.512
$c = 0.9$	0.227	0.165	0.415	0.417	0.383	0.422	0.386	0.412
$r = 40$								
$c = 0.3$	0.569	0.583	0.639	0.639	0.685	0.650	0.643	0.635
$c = 0.6$	0.380	0.312	0.513	0.511	0.540	0.522	0.511	0.509
$c = 0.9$	0.263	0.160	0.414	0.418	0.403	0.428	0.417	0.416
$r = 80$								
$c = 0.3$	0.601	0.611	0.642	0.639	0.694	0.655	0.657	0.639
$c = 0.6$	0.421	0.308	0.522	0.520	0.570	0.533	0.546	0.517
$c = 0.9$	0.299	0.161	0.415	0.416	0.434	0.430	0.443	0.415

Table 4: The means of the misclassification rate for Simulation 4 over 50 simulations. Most of the standard errors are less than 0.003, and the largest standard error is 0.005.

The misclassification rates are reported in Table 4. Comparing to the results for $d = 5$ in Table 3, we see that NDC-S, NC, NSC, LDA, and Logistic seem to be only minimally affected by the presence of irrelevant features. However, other classification methods, including NDC, KNN, and SVM, are all noticeably affected by the inclusion of irrelevant features, and their misclassification rates further increase as r increases. In particular, the difference between the behaviors of NDC and NDC-S in this setting demonstrates that by including feature selection as part of the algorithm, NDC-S becomes much more robust to the presence of irrelevant features. For datasets in the real world, it is often the case that some of the features are irrelevant, and therefore NDC-S might be a better default choice to use on real-world data.

To validate our hypothesis that the feature selection part of NDC-S is working as intended, we also compute the means and standard errors of the number of selected features for different classification methods, and the results are reported in Table 5. As we can see, NDC, NC, KNN, and SVM always use all the features. Comparing the feature selection of NDC-S, NSC, LDA, and Logistic, we could argue that in general NDC-S has the best performance. This is because for $c = 0.6$ and $c = 0.9$, regardless of the number of irrelevant features r , NDC-S always select close to 20 features, which is exactly the number of relevant features in the data.

	NDC	NDC-S	NC	NSC	KNN	LDA	SVM	Logistic
$r = 20$								
$c = 0.3$	40(0)	31(2)	40(0)	33(1)	40(0)	39(0)	40(0)	33(0)
$c = 0.6$	40(0)	23(1)	40(0)	29(1)	40(0)	38(0)	40(0)	34(0)
$c = 0.9$	40(0)	21(1)	40(0)	25(1)	40(0)	38(1)	40(0)	35(0)
$r = 40$								
$c = 0.3$	60(0)	40(3)	60(0)	44(2)	60(0)	58(0)	60(0)	42(1)
$c = 0.6$	60(0)	20(1)	60(0)	33(2)	60(0)	54(1)	60(0)	44(1)
$c = 0.9$	60(0)	20(0)	60(0)	27(2)	60(0)	52(1)	60(0)	45(1)
$r = 80$								
$c = 0.3$	100(0)	92(3)	100(0)	58(4)	100(0)	93(1)	100(0)	54(1)
$c = 0.6$	100(0)	20(0)	100(0)	41(4)	100(0)	84(1)	100(0)	56(1)
$c = 0.9$	100(0)	20(0)	100(0)	29(3)	100(0)	76(2)	100(0)	61(1)

Table 5: The means (and standard errors) of the number of selected features for Simulation 4 over 50 simulations.

8 Applications

In this section, we apply our method to three gene expression datasets, all of which were proposed and pre-processed by [de Souto et al. \[2008\]](#). In all three datasets, the rows represent different samples of tissues, and the columns represent different genes. The samples have already been labeled with different classes based on their types of tissue. We evaluate and compare the performance of the following eight classification methods: nearest disjoint centroid classifier without feature selection (NDC), nearest disjoint centroid classifier with feature selection (NDC-S), nearest centroid classifier (NC), nearest shrunken centroid classifier (NSC), k -nearest neighbors (KNN), penalized linear discriminant analysis (LDA), sparse support vector machine (SVM) [[Yi and Huang, 2017](#)], and logistic regression with L_1 regularization (Logistic). We perform 3-fold cross validation on the datasets, and report the means and standard errors of the misclassification rates.

In addition to misclassification rates, we also consider whether the classifiers can perform feature selection, and if yes, how many features are selected. We know that NDC, NC, and KNN require all the features to perform classification. However, a varying number of features can be selected by changing the threshold Δ in NSC, or changing the λ in NDC-S, LDA, SVM or Logistic. Those hyperparameters are selected by nested cross-validation to achieve the smallest misclassification rate on each fold, and we also report the means and the standard errors of the number of features selected by each classifier.

8.1 Breast and Colon Cancer Gene Expression Dataset

The first dataset consists of 104 samples and 182 genes. There are two types of samples: 62 samples correspond to breast cancer tissues, and 42 samples correspond to colon cancer tissues.

NDC	NDC-S	NC	NSC	KNN	LDA	SVM	Logistic
0.029(0.029)	0.029(0.017)	0.183(0.041)	0.115(0.028)	0.087(0.001)	0.048(0.009)	0.038(0.025)	0.029(0.016)

Table 6: The means (and standard errors) of the misclassification rate on the breast and colon cancer gene expression dataset.

The misclassification rates are reported in [Table 6](#). As we can see, NDC, NDC-S, and Logistic have the best performance, followed closely in turn by SVM, LDA, and KNN. Finally, NC and NSC have the worst performance, with misclassification rates over 11%. In particular, the small standard errors indicate that the difference between the performance of our method (NDC and NDC-S) and the two directly comparable classifiers (NC and NSC) is quite significant. One possible reason that our method performs well on this dataset is that there is a natural interpretation for the disjoint features that our method produced: two disjoint groups of genes that are useful for identifying breast and colon cancer, respectively. Since breast cancer and colon cancer are two completely different types of cancer, it is quite possible that the genes

that are useful in predicting one type of cancer are largely irrelevant to predicting another type of cancer. Therefore, our nearest disjoint centroid classifiers, which identify two disjoint sets of genes that are used in predicting the two types of cancer, perform better than the nearest centroid classifier and the nearest shrunken centroid classifier, both of which rely on the same set of genes to predict the two types of cancer and thus might incorporate more noisy and irrelevant information.

NDC	NDC-S	NC	NSC	KNN	LDA	SVM	Logistic
182(0)	83(7)	182(0)	87(48)	182(0)	125(19)	31(10)	11(1)

Table 7: The means (and standard errors) of the number of selected features on the breast and colon cancer gene expression dataset.

The number of selected features are reported in Table 7. For this dataset, Logistic only selects 11 features on average (among the three models built for the three folds), which is surprisingly small considering it achieves less than 3% misclassification rate with around 6% of the features. SVM also achieves less than 4% misclassification rate with around 17% of the features. However, comparing NDC-S and NSC, we see that NDC-S also achieve less than 3% misclassification rate while selecting 83 features on average, whereas NSC achieve more than 11% misclassification rate while selecting 87 features on average. In addition, compared to LDA, NDC-S uses few features and achieves a smaller misclassification rate.

8.2 Leukemia Gene Expression Dataset

The second dataset consists of 72 samples and 1868 genes. There are two types of samples: 47 samples correspond to acute myeloid leukemia, and 25 samples correspond to acute lymphoblastic leukemia.

NDC	NDC-S	NC	NSC	KNN	LDA	SVM	Logistic
0.056(0.014)	0.028(0.014)	0.056(0.028)	0.028(0.014)	0.153(0.077)	0.056(0.014)	0.083(0.024)	0.167(0.083)

Table 8: The means (and standard errors) of the misclassification rate on the leukemia gene expression dataset.

The misclassification rates are reported in Table 8. As we can see, all four centroid-based classification methods (NDC, NDC-S, NC, NSC) and LDA achieve misclassification rates that are less than 6%, whereas some other classification methods like KNN and Logistic have much larger misclassification rates.

NDC	NDC-S	NC	NSC	KNN	LDA	SVM	Logistic
1868(0)	57(13)	1868(0)	1868(0)	1868(0)	1710(6)	113(12)	10(3)

Table 9: The means (and standard errors) of the number of selected features on the leukemia gene expression dataset.

The number of selected features are reported in Table 9. For this dataset, it is worth noting that despite having equally good performance in terms of misclassification rates, NDC-S only selects 57 features on average (around 3% of the features), where NSC requires all the features. In addition, NDC-S outperforms both LDA and SVM by having a smaller misclassification rate and selecting fewer features. This is an example where only our NDC-S algorithm could give stellar performance in both classification and feature selection, whereas other competing classifiers could perform well in one aspect at most.

8.3 Breast Cancer Gene Expression Dataset

The third dataset consists of 49 samples and 1198 genes. There are two types of samples: 25 samples correspond to breast tumors that are estrogen-receptor-positive, and 24 samples correspond to breast tumors that are estrogen-receptor-negative.

The misclassification rates are reported in Table 10. As we can see, LDA achieves the smallest misclassification rate on this dataset, followed closely by NDC-S and other methods, although the difference between

NDC	NDC-S	NC	NSC	KNN	LDA	SVM	Logistic
0.183(0.032)	0.145(0.043)	0.206(0.057)	0.164(0.023)	0.186(0.064)	0.103(0.042)	0.164(0.023)	0.183(0.032)

Table 10: The means (and standard errors) of the misclassification rate on the breast cancer gene expression dataset.

the misclassification rates of most of the classifiers is not that significant after taking the standard errors into consideration.

NDC	NDC-S	NC	NSC	KNN	LDA	SVM	Logistic
1198(0)	15(3)	1198(0)	488(59)	1198(0)	846(91)	52(24)	7(3)

Table 11: The means (and standard errors) of the number of selected features on the breast cancer gene expression dataset.

The number of selected features are reported in Table 11. For this dataset, we notice that both NDC-S and Logistic are surprisingly efficient at identifying relevant features, selecting 15 and 7 features on average (around 1% of the features), respectively. In contrast, SVM, NSC, and LDA select around 4%, 41%, and 71% of features on average, respectively. This shows that our NDC-S algorithm is able to achieve the smallest misclassification rate with as few as 15 features (on average) out of 1198 features.

Name	Normalized Frequency	Description
X80062_at	0.96	SA mRNA
29610_s_at	0.80	GYPE Glycophorin E
X57129_at	0.66	HISTONE H1D
X02958_at	0.60	Interferon alpha gene IFN-alpha 6
X17025_at	0.57	Homolog of yeast IPP isomerase

Table 12: The top five most frequently selected genes in the breast cancer gene expression dataset.

Since the selected features are genes that might be biologically related to breast cancer, we decide to run the experiment 100 times and compute the normalized frequency of genes that get selected by our algorithm. The top five most frequently selected genes, their normalized frequencies, and their descriptions are listed in Table 12. Noticeably, the first two genes, named “X80062_at” and “29610_s_at”, get selected by our algorithm 96% and 80% of the time, respectively. This suggests that the biological relationship between these two genes and breast cancer might be worthy of further investigation.

9 Discussion

In this paper we developed a new classification method called the nearest disjoint centroid classifier. The two main differences between our nearest disjoint centroid classifier and the nearest centroid classifier are: (1) the centroids are defined based on disjoint subsets of features instead of all the features, and (2) the distance is induced by the dimensionality-normalized norm instead of the Euclidean norm. We presented and proved a few theoretical results regarding our method. In addition, we proposed a simple algorithm based on adapted k -means clustering that can find the disjoint subsets of features used in our method, and extended the algorithm to perform feature selection by making a few small adjustments. We evaluated and compared the performance of our method to other classifiers on both simulated data and real-world gene expression datasets. The results have demonstrated that in many situations, our nearest disjoint centroid classifier is able to outperform other competing classifiers by having smaller misclassification rates and/or using fewer features.

In the future, we plan to explore different ways of utilizing the disjoint subsets of features and the associated centroids obtained by our method. In this paper we focused on one simple and straightforward way to perform classification: classify a new data point to the class with the nearest disjoint centroid.

However, there are many other methods that could be adapted to using disjoint subsets of features instead of all the features. For example, we could fit a (multinomial) logistic regression model based on the distances from every data point to the k disjoint centroids. We could also define distances from a test data point to a training data point based on the training data point's class and the associated subset of features. Therefore, it is also possible to develop a version of the k -nearest neighbors algorithm with disjoint subsets of features.

Another interesting direction to pursue is to consider different ways to obtain the k subsets of features associated with the k classes. In this paper we used an adapted version of the k -means clustering algorithm to find those k subsets of features, which is simple but also restrictive: the k subsets of features must be disjoint. In general, our method could still work even if there is intersection between the k subsets of features. As a result, instead of performing k -way clustering on the features, we could consider performing two-way clustering on the features k times, each time obtaining one group of features for one class. In the end, we would obtain k groups of features that are not required to be disjoint. In addition, they are also not required to cover all the features, and the features that are not included in any of the k groups are not used in prediction. This means that it can also perform feature selection, although controlling the number of selected features would require additional work.

References

- M. Ajtai, J. Komlós, and G. Tusnády. On optimal matchings. *Combinatorica*, 4(4):259–264, 1984.
- L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. CRC press, 1984.
- C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
- C. Curtis, S. P. Shah, S.-F. Chin, G. Turashvili, O. M. Rueda, M. J. Dunning, D. Speed, A. G. Lynch, S. Samarajiwa, Y. Yuan, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012.
- A. R. Dabney. Classification of microarrays to nearest centroids. *Bioinformatics*, 21(22):4148–4154, 2005.
- A. R. Dabney and J. D. Storey. Optimality driven nearest centroid classification from genomic data. *PLoS One*, 2(10):e1002, 2007.
- M. C. de Souto, I. G. Costa, D. S. de Araujo, T. B. Ludermir, and A. Schliep. Clustering cancer gene expression data: a comparative study. *BMC Bioinformatics*, 9(1):497, 2008.
- V. Dobrić and J. E. Yukich. Asymptotics for transportation cost in high dimensions. *Journal of Theoretical Probability*, 8(1):97–118, 1995.
- R. M. Dudley. The speed of mean glivenko-cantelli convergence. *The Annals of Mathematical Statistics*, 40(1):40–50, 1969.
- R. Edgar, M. Domrachev, and A. E. Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002.
- N. Fournier and A. Guillin. On the rate of convergence in wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3):707–738, 2015.
- N. Fraiman and Z. Li. Biclustering with alternating k-means. *preprint arXiv:2009.04550*, 2020.
- J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

- E.-H. S. Han and G. Karypis. Centroid-based document classification: Analysis and experimental results. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 424–431. Springer, 2000.
- D. J. Hand and K. Yu. Idiot’s bayes—not so stupid after all? *International Statistical Review*, 69(3):385–398, 2001.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data mining, Inference, and Prediction*. Springer Science & Business Media, 2009.
- J. Horowitz and R. L. Karandikar. Mean rates of convergence of empirical measures in the wasserstein metric. *Journal of Computational and Applied Mathematics*, 55(3):261–273, 1994.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012.
- V. Lertnattee and T. Theeramunkong. Effect of term distributions on centroid-based text categorization. *Information Sciences*, 158:89–115, 2004.
- I. Levner. Feature selection and nearest centroid classification for protein mass spectrometry. *BMC Bioinformatics*, 6(1):1–14, 2005.
- D. D. Lewis, Y. Yang, T. Russell-Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.
- J. S. Parker, M. Mullins, M. C. Cheang, S. Leung, D. Voduc, T. Vickery, S. Davies, C. Fauron, X. He, Z. Hu, et al. Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of Clinical Oncology*, 27(8):1160, 2009.
- S. Rachev. *Probability Metrics and the Stability of Stochastic Models*. Wiley Series in Probability and Statistics - Applied Probability and Statistics Section. Wiley, 1991.
- S. T. Rachev and L. Rüschendorf. *Mass Transportation Problems: Volume I: Theory*, volume 1. Springer Science & Business Media, 1998.
- O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- T. Sørlie, R. Tibshirani, J. Parker, T. Hastie, J. S. Marron, A. Nobel, S. Deng, H. Johnsen, R. Pesich, S. Geisler, et al. Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proceedings of the National Academy of Sciences*, 100(14):8418–8423, 2003.
- S. Tan. An improved centroid classifier for text categorization. *Expert Systems with Applications*, 35(1-2): 279–285, 2008.
- R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572, 2002.
- R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science*, pages 104–117, 2003.
- S. Volinia, G. A. Calin, C.-G. Liu, S. Ambs, A. Cimmino, F. Petrocca, R. Visone, M. Iorio, C. Roldo, M. Ferracin, et al. A microrna expression signature of human solid tumors defines cancer gene targets. *Proceedings of the National Academy of Sciences*, 103(7):2257–2261, 2006.
- J. Weed and F. Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *Bernoulli*, 25(4A):2620 – 2648, 2019.
- D. M. Witten and R. Tibshirani. Penalized classification using fisher’s linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):753–772, 2011.

C. Yi and J. Huang. Semismooth newton coordinate descent algorithm for elastic-net penalized huber loss regression and quantile regression. *Journal of Computational and Graphical Statistics*, 26(3):547–557, 2017.